INDEKS Nr 32034 X ISSN 1230-7947

Commodore

3/1994

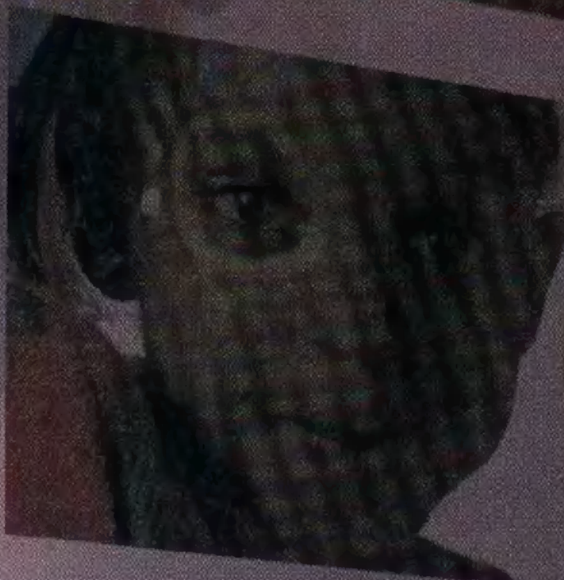
AMIGOWIEC

Pismo użytkowników komputerów AMIGA cena 25.000 zł

PICCOLO

Oprogramowanie:

- REAL 3D v1.4
- MAXON CINEMA
- SCALA MM300



WARSZTATY!

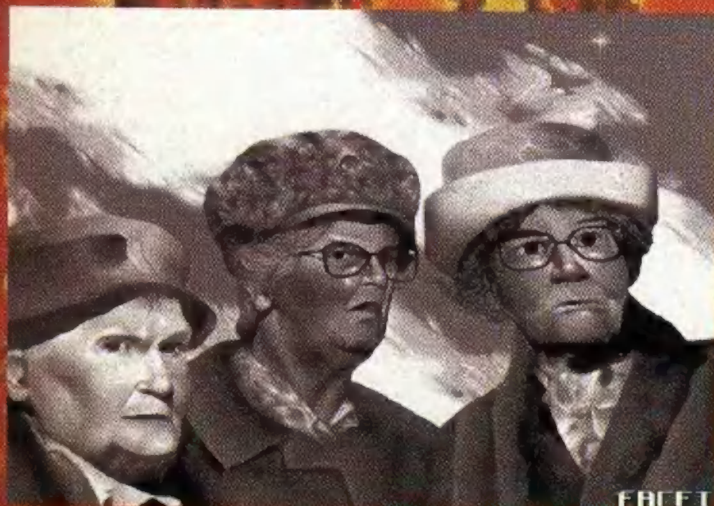
- AMOS
- Cross DOS
- Arexx cz. 4
- Imagine cz. 3
- Protracker
- Modem

Ciąg dalszy na stronie 75








FACET






TRAJAN


Claudia
Schiffer





AMIGOWIEC

ROK 5 NR 3/94 (37) INDEX 32034X

REDAKCJA

REDAKTOR NACZELNY:
RYSZARD KOWALSKI

SEKRETARZ REDAKCJI
TOMASZ ŁOBODA

Z-CA REDAKTORA NACZELNEGO:
KRZYSZTOF NOWICKI

OPRACOWANIE GRAFICZNE:
KRZYSZTOF WIRSZYŁŁO
BARTŁOMIEJ WRÓBLEWSKI

SKŁAD KOMPUTEROWY:
MAŁGORZATA LEWANDOWSKA

KOREKTA
KAROL JAGODZIŃSKI
ŁUCJA UMIŃSKA

KOLPORTAŻ:
ANDRZEJ KENTZER

PRENUMERATA:
AGNIESZKA PRZYBYLSKA

WSPÓŁPRACA:
WOJCIECH BIAŁKOWSKI,
WOJCIECH CZYZ,
JAROSŁAW CHROSTOWSKI,
MARCIN GACKOWSKI,
ADAM GREGROWICZ,
TOMASZ FLANC,
TOMASZ HRYCUNIAK,
MACIEJ KLIMKIEWICZ,
TOMASZ KOKOSZCZYŃSKI,
SEBASTIAN KŁOMSKI,
TOMASZ KULBACKI,
TOMASZ ŁOBODA,
ARTUR ŁUKASIK,
TOMASZ MATAJEK,
ROBERT RUTKOWSKI,
IZABELA SKIBIŃSKA,
MAREK STOR,
KRZYSZTOF TROJANOWSKI,

ADRES REDAKCJI:
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. (+52) 28-79-20, fax (+52) 22-64-03

KONTO BANK PKO S.A. BYDGOSZCZ
00509011-04009539-2611-30-00111o

SKŁAD NA KOMPUTERACH AMIGA
OKŁADKA: WOJCIECH BIAŁKOWSKI
DRUK: "ZAKŁADY GRAFICZNE" SP. Z O.O
UL. OKRZEI 5, 64-920 PIŁA

Nowy image?



Witam Was w trzecim numerze Amigowca. Na wstępie chciałbym podziękować Wam za wszystkie listy. Czytamy je uważnie i notujemy uwagi. Niestety muszę z przykrością poinformować, że ani ja, ani moi koledzy z redakcji nie jesteśmy w stanie odpisywać prywatnie na poszczególne listy. Nie miejcie za to do nas żalu. Wszystkie Wasze problemy postaramy się omówić w Pismach, Pisemkach lub w odpowiednim Warsztacie.

Z zadowoleniem odnotowaliśmy spore zainteresowanie numerami archiwalnymi. W chwili obecnej dostępne są tylko numery zaznaczone na blankietach prenumeraty. W tym miejscu ukłon w kierunku tych, których zmyliły różne ceny numerów archiwalnych oraz prenumeraty zamieszczone w dwóch poprzednich numerach Amigowca. Dla czytelników, którzy chcieliby nabyć pozostałe, niedostępne u nas archiwalia, pozostaje tylko Ami-Market.

Ze względów ekonomicznych musieliśmy zrezygnować z konkursu okładkowego. Teraz kupon jest już tylko zamówieniem. My z naszej strony postaramy się uaktualniać zestawy i konstruować je tak, aby były bardziej atrakcyjne. Laureatami konkursu okładkowego nr 2 zostali:

Ernest Noga - Masłów woj. kieleckie
Filip Wośko - Bolesławiec
Radosław Leśkiewicz - Łuków woj. siedleckie

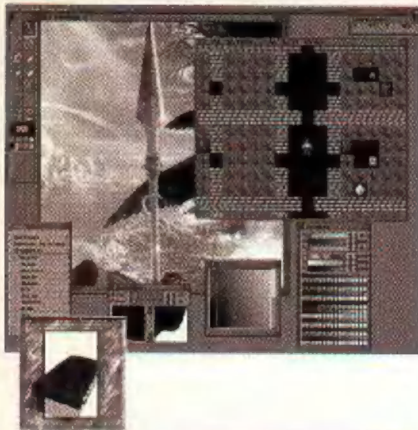
Gry prześlemy pocztą.

W listach poruszacie temat Amigi w szkołach, a raczej niechęci (niewiedzy!?) dyrektorów do tego komputera. Jakie są przyczyny takiego zachowania? Jak słusznie zauważył pan Zbigniew Zieliński z Poznania IBM dominuje wszędzie, począwszy od sklepu z parówkami, na ośrodkach obliczeniowych skończywszy. Wobec powszechnie utartego (nie bez pomocy Commodore) przekonania, że Amiga jest komputerem do zabawy (co nie jest prawdą, gdyż znam kilka osób, które wydały ok. 100 mln. zł na IBMa i teraz grają w Strike Commandera z płynną animacją), a IBM sprzętem "profesjonalnym" do pracy, taka reakcja decydentów wcale mnie nie dziwi. W normalnych warunkach zmiana takiego podejścia należy do przedstawicielstwa firmy, które przez działania reklamowe może uświadamiać i przekonywać przyszłych potencjalnych nabywców swojego sprzętu. A Commodore co? Likwiduje przedstawicielstwo, a sprawy promocji pozostawia swoim dealerom. Ci z kolei są zajęci robieniem "bussinesu" w polskim tego słowa znaczeniu, czyli sprzedażą jak największej ilości Amig 1200 i monitorów, zapominając o ofercie sprzętu wyższej klasy, czy o jakiejkolwiek promocji. Bo i po co? To jest przecież zadanie przedstawicielstwa, a i pieniądze trzeba wydać. I kółeczko się zamyka? Jedyne co można zrobić w tej sprawie to pisać, mówić i przekonywać. Samemu. Prywatnie. I chwała wszystkim tym, którym uda(ło) się przekonać innych o swojej racji. My ze swojej strony kiedy tylko można i gdzie tylko można staramy się promować naszą przyjaciółkę. Wstyd przyznać, ale to właśnie dzięki ludziom skupionym wokół pism amigowych i licznemu gronu entuzjastów, ten komputer nadal istnieje i ma się dobrze. Ostatnie poczynania Commodore pozwalają mieć cichą nadzieję, że w przyszłości coś zmieni się na lepsze.

My tym czasem wracamy do siebie. W dzisiejszym numerze troszkę większa dawka informacji na temat grafiki komputerowej, a w szczególności ray-tracingu. Szczególnie polecam pozycje Imagine 2.9 oraz Maxon Cinema, a także bardzo ciekawy artykuł Imagine cz. 3 z działu Warsztaty Grafika.

Do zobaczenia.

R. Kowalski



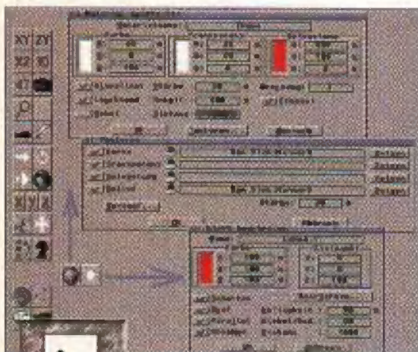
7 PICOLLO

Kolejna 24-bitowa karta graficzna systemu EGS. Co nowego w systemie EGS? Przeczytajcie!



11 KOPROCESOR MATEMATYCZNY

Wojtek Czyż prezentuje zasady działania koporocesora matematycznego i sposoby jego programowania.



19 MAXON CINEMA

Tym razem firma Maxon przedstawiła program do ray-trackingu. Jakie są jego wady i zalety opowie Tomasz Łoboda.

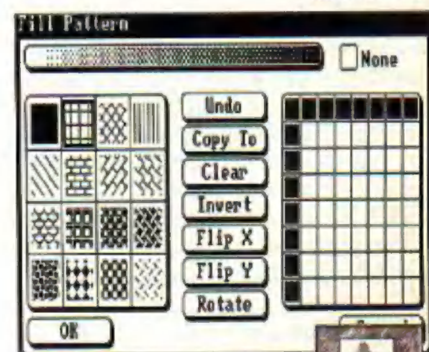


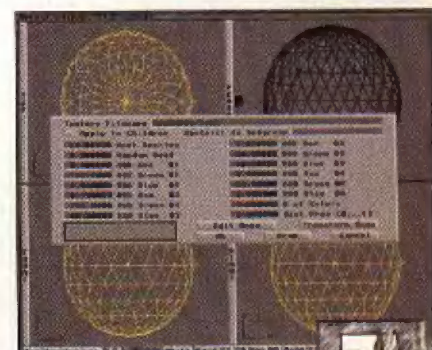
22 SCALA MM 300

Co nowego słychać w Scali? Jakie zmiany wprowadzono w najnowszej wersji?


IMAGINE
Version 2.9
UNDER CONSTRUCTION


FLYING


Fill Pattern
None
Undo
Copy To
Clear
Invert
Flip X
Flip Y
Rotate
OK





Commodore na CeBicie

W dniach 16-23 marca odbyły się w Niemczech targi CeBit '94. Oczywiście zaszczyciła je swoją obecnością również firma Commodore. Główną wagę położono na prezentację Amigi CD32 oraz Amigi 4000.

Na targach można się było zapoznać z aktualną ofertą firmy - od A... miga do Z... obacz sam.

Coraz więcej filmów jest tworzonych przy użyciu Amig. Tu już nie chodzi tylko o serial "seaQuest" lecący aktualnie w programie RTL, o którym wspominaliśmy w poprzednim Amigowcu. Kolejne tytuły to "Babylon 5", "Deep Space Nine", "Tom Knockers", czy "Quantum Leap". Oczywiście Amiga nie wypiera aktorów - pozwala jedynie na tworzenie efektów specjalnych. Na targach można było zasięgnąć dalszych informacji o serialu "seaQuest" udzielanych przez system Touch-Screen (Dotknij-Ekranu), obsługiwany przez Amigi. Pokazywano również tworzenie poszczególnych scen trikowych.

Oczywiście Amiga 4000 nie jest tylko komputerem do tworzenia "jakichś tam" trików. Przy jej pomocy z zastosowaniem najnowszych wersji programów Scala, czy MediaPoint można dokonywać całkiem profesjonalnych prezentacji w dziedzinie biznesu, czy nauki. Dodatkowym plusem Amigi 4000 jest to, że od momentu pokazania na targach będzie ona dostępna w wersji Tower z procesorem 68040, 6MB pamięci i wewnętrznym dyskiem CD-ROM.

Przebojem amigowskiej części targów była oczywiście Amiga CD32 z modułem MPEG, który umożliwia szybkie rozpakowywanie mocno spakowanych danych np. video. System ten jest określany inaczej jako Full Motion Video i przy jego pomocy można obejrzeć 74 minuty filmu z jednego dysku CD. Na razie dostępne są jedynie "Top Gun", "Star Trek 6" i "Phantom of the Opera", ale z pewnością wkrótce oferta znacznie się poszerzy.

Na targach nie zabrakło oczywiście Amigi 1200. Ostatnio jest ona uznawana jako domowy system obróbki video. Wyposażona w Genlock i specjalne oprogramowanie umożliwia amatorom videofilmowania profesjonalną obróbkę samodzielnie nakręconych sekwencji.

Piraci w odwrocie!

Jak wiemy piractwo jest zmorą nie tylko polską. Kłopoczą się nim również państwa o bardziej uregulowanych systemach prawnych niż nasz. Raz wychodzi im to lepiej, innym razem gorzej. Ostatnio angielska organizacja o dźwięcznej nazwie FAST (The Federation Against Software Theft) zadała miażdżący cios jednemu z popularniejszych BBSów o miłej dla ucha nazwie Arcadia.

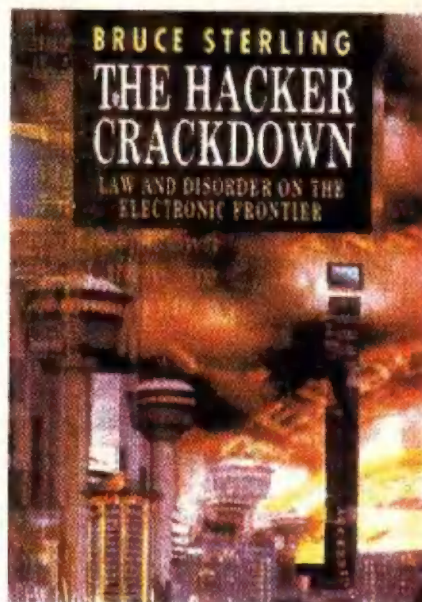
BBSy służą do wymiany danych przez telefon, bez konieczności siedzenia przy aparacie. Są to inaczej takie stacje pocztowe dla komputerowców z modemami. Również dzięki nim możliwa jest wymiana nielegalnych informacji - jak np. odbezpieczonych kopii co lepszych programów. Oczywiście nie wszyscy mają łatwy dostęp do takich żył złota. Każdy "specjalny" użytkownik ma swoje hasło i tylko dzięki niemu może "pobierać" smakowite kąski. Użytkownikami są oczywiście piraci w innych rejonach danego kraju, czy nawet w innych krajach.

Zgodnie zresztą z hasłem: "Piraci całego świata łączcie się".

Tym razem FAST wspólnie z oficerami jednostki specjalnej do spraw komputerów Scotland Yardu wpadli na adres Arcadii. System bazował na Amigach (a niektórzy mówią, że na tym sprzęcie nie da się stworzyć profesjonalnego BBSu) i był wyposażony w 5 szybkich modemów oraz w setki dyskietek komputerowych, że o dyskach twardych nie wspomnę. Arcadia była jednym z największych BBSów działających w Wielkiej Brytanii i przez nią przechodziła cała masa nielegalnego oprogramowania z całego świata. Trwało to ponad trzy lata. Skasowanie tego BBSu wywołało panikę pośród innych pirackich BBSów. Podobno wiele z nich zniknęło tej samej nocy...

Na piratach też można zarobić!

Zarabiają nie tylko piraci. Zarabia się również na nich. Przykładem tego jest książka wydana ostatnio w Anglii, poświęcona historii piractwa od mniej więcej roku 1990 - "Hacker Crackdown". Książka ta jest wspaniałą lekturą dla wszystkich entuzjastów cyberpunka. Również ci, którzy są przeciwni idei hackerstwa (nie mylić z harcerstwem) znajdą w niej wiele pożytecznych informacji, o tym jak w innych krajach radzą sobie z tym problemem.



Nowa, rasowa oferta Commodore

Firma Commodore wzięła się nieco do roboty i oprócz wypuszczania na rynek Amig 1200 i 4000 postanowiła je sprzedawać w większej ilości. W celu zachęcenia klientów do kupna półprofesjonalnego komputera jakim jest Amiga 1200 na rynku pojawił się specjalny zestaw o dźwięcznie brzmiącej nazwie: "Desktop-Dynamite". Składa się on z Amigi 1200 w wersji podstawowej, 3 programów użytkowych i 2 gier.

Obróbką tekstu zajmuje się w tym zestawie program "Digita Wordworth", wspomagany przez drugi program z zestawu: "Digita Print Manager", który ma zapewnić bezbłędny wydruk całości. Program do obróbki tekstów jest podobny do FinalWritera i w wersji ostatecznej ma posiadać wiele zalet tego





ostatniego. Wspomaganie go programem drukującym być może poprawi jakość wydruku na zwykłych drukarkach. Trzeci z programów użytkowych to znany nam wszystkim Deluxe Paint IV w wersji działającej na układach graficznych AGA. Przedstawiać chyba nie muszę.

Gry, to "Dennis" o nazwie jak tytuł filmu, z którego wzięła się ta historyjka o dzielnym panu Wilsonie, oraz "Oscar" - prosta gra przygodowa, w której chodzi o zdobycie złotego Oscara za najlepszy film.

Czy ta oferta chwyci? Różne są gusta na niemieckim rynku, bo tam jest ona kierowana, ale wydaje się jednak, że całość może pociągnąć do boju DPaint IV. Oby nie za późno. Cena zestawu nie została podana.

Nowe wirusy w ataku

Projektanci wirusów niestety nadal nie mogą spać po nocach i robią wszystko, żeby umilić nam życie. Ciągle pojawiają się nowe wirusy i ciągle musimy czekać na nowe programy odwszawiające.



Wracając do najnowszych wirusów, to wcale nie jest powiedziane, że np. Virus_Checkerowi uda się je zwalczyć. Przynajmniej nie temu z wersją 6.33.

Creeping Evel to miły wirus bootblockowy, którego działalność przejawia się wyświetleniem niemieckiej flagi i zniszczeniem cylindrów startowych dyskietek i dysku twardego. Milusiński. Przy odpalaniu wszelakich programów z dyskietek (tak, że wykorzystywany jest ich bootblock) radzę wyłączać twardy dysk.

Innym milusińskim jest koń trojański Descriptor 3.0 Trojan. Wirus ten jest wirusem plikowym i rozpowszechniany jest w dwóch postaciach: jako Descr4.0.lha, który ma powodować, że program PowerSnap działa szybciej oraz jako z-speed.lha, który po rozpakowaniu okazuje się przyspieszaczem modemów Zenith (Zenith Modem Speeder). Oba te programy powodują uruchomienie Descriptora 3.0, który szuka w katalogu s: pliku o nazwie "descriptions.txt". Jeśli go nie znajdzie to zawiesza system, natomiast jeśli na nasze nieszczęście mamy tam taki plik, to możemy pożegnać się z zawartością twardego dysku, co przy obecnych technikach robienia backupu może być fatalne w skutkach. Obecnie jedynym programem, który wyłapuje te wirusy jest VirusZ.

Dodatek do Deluxe Painta

Deluxe Paint, jeden z popularniejszych programów graficznych na Amigę i w ogóle, wcale nie jest taki łatwy w obsłudze, jak to się może wydawać na pierwszy rzut oka. Zawiera wiele funkcji, które można zacząć wykorzystywać dopiero po dłuższym kontakcie z programem. Wiele z nich daje się wywołać przy pomocy różnego rodzaju skrótów klawiaturowych. O ile zapamiętanie tych skrótów jest w miarę łatwe przy korzystaniu z edytorów tekstu (praca z klawiaturą), o tyle może nastręczać pewne trudności grafikom. Obsługa programu z klawiatury uzupełnia obsługę myszą i niejednokrotnie może prowadzić do znacznego przyspieszenia pracy. Wbrew temu, co twierdzą niektórzy, że przy każdym programie trzeba by się na nowo uczyć klawiatury, uważam, że tych programów, których naprawdę często używamy wcale nie ma tak wiele i warto zapamiętać klawiaturowe wywołania ich niektórych funkcji.



Amerykańska firma A&P Software pomyślała o tych wszystkich zagubionych grafikach próbujących się tego nauczyć i wypuściła na rynek plastikową nakładkę na Amigę. Wypisano na niej większość skrótów z jakich można skorzystać przy pracy z Deluxe Paintem. Nakładka ta pasuje praktycznie do wszystkich Amig (z wyjątkiem oczywiście biednej 600-tki), ale kosztuje niestety 10$. Cena "dosyć" wysoka zważywszy nasze zarobki. A może tak rodzimi producenci pomyśleliby o czymś takim? Panowie, pieniądze leżą na ziemi...

Proste podróże po Niemczech.

Wszystkim, którzy dużo podróżują po Niemczech można śmiało polecić programik "Streckenplaner". Program ten jest świetną pomocą w planowaniu Waszych podróży po RFN. Wystaczy podać miejscowość, w której się znajdujemy oraz miejsce gdzie chcemy dojechać, a programik załatwi za nas znalezienie najkrótszej trasy. Nie trzeba męczyć się z wielkimi mapami i szukać po atlasach. Propozycje trasy ukazują się na ekranie w postaci albo wektorowej mapy, którą możemy dowolnie powiększać, albo jako prosty opis drogi w trzech stopniach komplikacji. Oczywiście wszystko można bez problemu przerzucić na drukarkę. W program wepchnięto wszystkie miejscowości w RFN liczące sobie powyżej 15.000 mieszkańców. Obsługa jest prosta (o ile ktoś zna język niemiecki), a program działa na wszystkich systemach.

Amigowa pornografia

Seks jest taką samą dziedziną życia jak np. czytanie książek. Tylko tak jakoś się dziwnie składa, że wywołuje "nieco" większe emocje. Nie ominął on również świata komputerów. Bo to przecież nic prostszego jak zdigitalizować jakiś "ładny" obrazek i puścić go w obieg. Obrazki mogą być różne. Najciekawsze są oczywiście te zakazane. Na dysku jest je o wiele łatwiej przenosić. No bo o ile gazetę pod poduszką można łatwo wykryć, o tyle do dysku ma dostęp tylko nasza pociecha, która się na komputerze zna. Reszta rodziny tkwi w błogiej nieświadomości co do zawartości dyskietki. Wszystko cacy, a hipokryzja kwitnie.

"Co robisz?"

"Ee nic, bawię się z komputerem."





Tylko, że ten komputer może pokazywać różne rzeczy. Przy obecnej jakości grafiki rozebrane panie wyglądają całkiem ładnie. Żeby to chodziło tylko o zdjęcia. Wiele firm zajmuje się tworzeniem specjalnych gier, w których chodzi o to by pan "dołożył" pani. Rozrywka raczej mało wyszukana, ale trafiająca też na mało wyrobione gusta naszych milusińskich. Wiele gier niby to logicznych jak Poker, Reversi ma w "podtekście" rozebrane panienki, które rozbieramy jeszcze bardziej wygrywając "z nimi" partyjkę. W Polsce, gdzie w żaden sposób nie daje się skontrolować obrotu nielegalnym oprogramowaniem, nie może być nawet mowy o kontroli "legalnych" obrazków rodem z pornokomputerografii przenoszonych na dyskietkach komputerowych. O tym aspekcie jakoś dziwnie się zapomina i rodzice wydają się wymagać uświadomienia, że wraz z zakupem komputera wkraczają do naszych domów rozebrane panienki. Nie kupować? Oczywiście nie jest to żadne rozwiązanie. Po prostu nie bądźmy hipokrytami.

Poradnik kochanków

Ostatnio firma Supervision wypuściła na rynek "Lover's Guide" - poradnik kochanków. Są tam nie tylko odpowiednie sceny, ale dzięki komputerowi mamy szybszy dostęp do odpowiednich "opcji". Na dysku znajdują się fragmenty dotyczące komunikacji w dziedzinie seksu, uprawiania miłości dla przyjemności i zabawy, metody zabezpieczania się przed niepożądaną ciążą i instrukcja prowadzenia gier wstępnych. Pytania i odpowiedzi. Jest tam wszystko. No może oprócz ducha miłości. Zamiast niego jest banalność. Niekoniecznie sprowokowana przez komputery...



Kontroler CD dla Amigi 1200 i 600

Od dłuższego czasu na rynku były dostępne kontrolery umożliwiające współpracę zwykłej 500-tki z CD-ROMem. Wystarczyło z boku wepchnąć odpowiedni kawałek hardware'u i wszystko grało. Na taki stan rzeczy zazdrośnie spoglądali posiadacze Amig 600 i 1200, których przyjaciółki były zaopatrzone w innego typu boczne otworki - tzw. PCMCIe. Ostatnio firma BSC zapowiedziała wypuszczenie na rynek kontrolera CD-ROMu, który będzie pasował właśnie do tej PCMCI. Kontroler współpracuje ze stacjami CD-ROM firmy Mitsumi o numerach LU005 i FX001D. Instalacja pakietu następuje oczywiście automatycnie i jest zgodna z systemem zapisu dysków ISO 9660. System jest kombatybilny z systemem CD-FileSystem umieszczonym na Workbenchu 3.1 oraz z systemami BabelCDFS, ASIM CD-ROM i Photo-CD. Sprzęt wymaga Kickstartu 2.04 lub wyższego.

Nowe, szybsze emulatory Vortexa

Firma Vortex słynie ze swoich kart Golden Gate emulujących IBMa na Amidze. Karty te budzą zaufanie wielu amigowców zmuszonych do korzystania z IBMków. Ostatnio pojawiły się na rynku karty jeszcze szybsze: 486SLC taktowana zegarem 25MHz i najnowsza: 486SLC2 taktowana zegarem 50MHz wewnętrznie i 25MHz zewnętrznie. Współczynnik szybkości Norton SI osiąga przy pierwszej z nich wartość 43, zaś przy drugiej 85. Oba procesory są kompatybilne z zestawem komend 486SX i wykorzystują 32-bitową szynę wewnętrzną i 16-bitową zewnętrzną. Oba też posiadają 1 KB pamięci notatnikowej. Karty te są odpowiednio 2.4 i 4 razy szybsze od procesora 386SX z taką samą częstotliwością zegara.

Zmieniona została również konfiguracja pamięci RAM. Od teraz wszystkie karty Golden Gate wyposażone są w 2.5 MB RAMu. Ceny kart niestety są wysokie: 948 DM za pierwszą i 1298 DM za drugą. Oprócz zmian sprzętowych wprowadzono zmiany w oprogramowaniu dostarczanym wraz z kartami. Wprowadzono do nich tzw. FAST OPTION, która poprawia szybkość o 20%. Możliwa jest również wymiana starszego oprogramowania w kartach Golden Gate 386SX. Wystarczy przesłać dyskietkę i opłacić porto.

GVP CeBit

GVP jest największym producentem peryferiów do Amigi. Także i w tym



roku firma zaprezentowała na targach Cebit swoje nowe produkty.

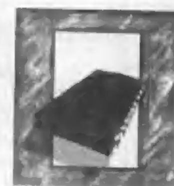
Długo oczekiwana karta turbo do Amigi 4000. Jej osiągi są ok. 4 razy lepsze niż Amigi 4000. Karta jest wyposażona w procesor MC 68040 40 MHz i może być opcjonalnie rozszerzona do 128 MB pamięci RAM za pomocą specjalnej karty dołączanej do karty turbo. Inną zaletą karty jest możliwość dołączenia szybkiego kontrolera SCSI II. Wszystkie opcjonalne rozszerzenia karty, zarówno RAM, jak i kontroler SCSI, nie zajmują ani jednego slotu ZORRO!!!

Firma nadal promuje swoją kartę graficzną EGS-Spectrum. Karta działa w systemie EGS RTG, który staje się coraz bardziej popularny wśród użytkowników kart graficznych (istnieje już kilka kart działających w tym systemie). Po dokładne informacje na temat karty SPECTRUM zapraszamy do Amigowca 1/94.

EGS-SpectraPaint to najnowsza wersja programu malarskiego dołączanego do kart Spectrum. GVP zaprezentowała ponadto długo oczekiwaną wersję ImageFX w pełni wspierającą system EGS RTG. Informacja ta z pewnością ucieszy wielu użytkowników karty, dla których użytkowanie dotychczasowej wersji programu nie było możliwe.

Nowy Final

Jak nas poinformowała firma SoftWood w chwili obecnej prowadzone są prace nad nową wersją Final Writera. Final Writer - Release 2 będzie rozwinięciem wersji poprzedniej. Zostaną usunięte wszystkie wykryte błędy, a także dodane nowe opcje m.in. UNDO. Czekamy z niecierpliwością. ❑



PICCOLO

Tomasz Kulbacki

W styczniowym numerze Amigowca mieliśmy okazję zapoznać się z doskonałą kartą graficzną EGS 28/24 Spectrum produkcji amerykańskiej firmy GVP. Pozytywna opinia T. Matajka - autora opisu ww. karty oraz własne kilkumiesięczne doświadczenia związane z jej użytkowaniem skłoniły nas do do bliższego zainteresowania się produktami innych producentów, oferujących karty graficzne zgodne z Enhanced Graphic System.

Długotrwałe wertowanie zachodnich czasopism naprowadziło nas w końcu na ślad niemieckiej firmy Ingenieurbüro Helfrich produkującej dwie karty graficzne zgodne z EGS i porównywalne z produktami GVP. Na zaznajomienie Czytelników z profesjonalną kartą EGS RAINBOW III spróbujemy znaleźć nieco miejsca w następnych numerach Amigowca, a tymczasem zajmiemy się jej mniejszym bratem - EGS Piccolo.

Piccolo jest bardzo efektywną kartą graficzną do Amigi 2000/3000/4000 umożliwiającą poprzez zastosowanie nowoczesnego kontrolera grafiki dowolne programowanie rozdzielczości (320*200 do 1600*1280 pikseli) oraz palety barw (2 do 16777216 kolorów). Dane te dotyczą wersji karty z 2 MB Video RAM. Kontroler graficzny posiada blitter osiągający prędkość transmisji danych do 30 MB/s. Automatyczne rozpoznawanie złącza ZORRO III (z Super Busterem v11) zapewnia optymalne wykorzystanie możliwości, jakie stwarza 32-bitowa szyna danych. Karta wyposażona jest także w sprzętowy kursor o maksymalnych wymiarach 64*64 piksele. Krótko mówiąc podstawowe dane techniczne Piccolo są identyczne z danymi karty EGS Spectrum (GVP). Zainteresowani dokładniejszymi informacjami znajdą je w numerze 1/94 Amigowca.

Piccolo w odróżnieniu od EGS Spectrum charakteryzuje się wieloma dodatkowymi atrakcjami. Karta posiada na płycie lokalną szynę video, do której można podłączyć dodatkowe moduły (na razie w fazie prototypów). Firma zapowiada wypuszczenie w pierwszym kwartale b.r. modułu umożliwiającego przetwarzenie sygnału RGB na CVBS i Y/C, digitizera funkcjonującego w czasie rzeczywistym z zaimplementowaną sprzętowo kompresją obrazu (JPEG) oraz karty składającej się z digitizera i modułu służącego do odtwarzania obrazu video w technice PIP (Picture in Picture). Piccolo wyraźnie dystansuje produkt GVP, jeśli porównamy oprogramowanie dostarczane w standardowych pakietach obu kart. Same biblioteki EGS nie różnią się oczywiście niczym, jednak przewaga Piccolo ujawnia się przy przeglądaniu i porównywaniu programów narzędziowych. Przede wszystkim jest ich więcej. Ingenieurbüro Helfrich dołącza do swej karty wiele użytecznych narzędzi, takich jak:

PicoRetarget

Program umożliwiający uruchomienie na karcie takich programów, które nie posiadają możliwości definiowania ekranu na jakim mają się uruchomić, np. PageStream 2.22 (programy te otwierają tzw. Custom Screen),

EGS-Mine

Dobry stary Boulder Dash tylko, że w większej ilości kolorów,

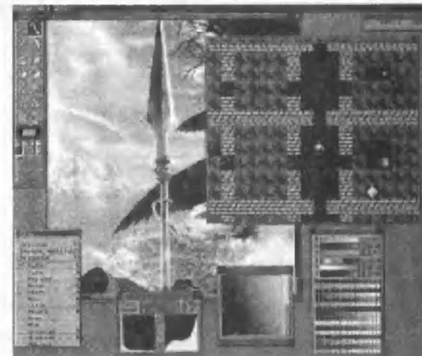
TV-Paint Junior

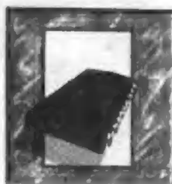
Nie wymaga wyjaśnień.

PicoPaint

Ten program graficzny ma wprawdzie swojego odpowiednika w konkurencyjnym produkcie zza oceanu w postaci programu EGS-Paint, ale naszym zdaniem Pico wyraźnie góruje nad produktem GVP większą ilością narzędzi (np. smear, regulowany smooth, definiowany spray).

Piccolo nie stawia użytkownikowi wysokich wymagań sprzętowych. Do





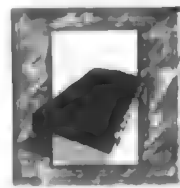
uruchomienia karty wystarcza Amiga z procesorem MC 68000, 2MB RAM, Kickstartem 2.04 i dowolnym dyskiem twardym. Ze względu na możliwość dowolnego programowania sygnału wyjściowego karty, można do niej podłączyć dowolny monitor RGB. Taka minimalna konfiguracja nie jest jednak w stanie udostępnić użytkownikowi wszystkich możliwości drzemiących w Piccolo. Producent zaleca jako optymalny zestaw:

Amiga z procesorem >= 68030
Pamięć >= 4 MB
Monitor Multiscan do 48 kHz
Kickstart 2.x, lub 3.x

Na koniec przedstawiamy test wykonany programem IntuiSpeed służącym do testowania kart graficznych. Porównywaliśmy prędkość karty z prędkością standardowych kości AGA w rozdzielczościach 640*480 i 256 kolorach. Trzeba tu jeszcze dodać, że używanie karty Piccolo w ilości kolorów mniejszej od 256 mija się z celem, gdyż w odróżnieniu od kości Amigi, zmniejszenie ich liczby nie wpływa na zwiększenie szybkości wyświetlania, a wręcz ją spowalnia! W związku z tym przedstawiamy również wyniki pracy kości AGA w 16 kolorach. Wartości większe stanowią o większej szybkości. Wyniki tego testu świadczą same za siebie, więc pozostawiamy je bez komentarza. ❑

Tabelka: Porównanie prędkości karty z prędkością standardowych kości AGA w rozdzielczościach 640*480 i 256 kolorach.

Rozdzielczość (sterownik)	PICOa:VGA 640x480 w 256 kolorach	MULTISCAN:Productivity w 256 kolorach	MULTISCAN:Productivity w 16 kolorach
Rysowanie punktów	245232	124639	237335
Rysowanie linii	36580	1491	5505
Zamalowywanie powierzchni	18675	495	1764
Przewijanie pionowe	1503	27	157
Przewijanie poziome	1162	28	144
Rysowanie okręgów	2009	633	2041
Tekst bez przewijania	9721	1825	6290
Rysowanie ramek	16114	692	2563
Zamykanie i otwieranie okien	46	21	49
Zmiana wielkości okna	119	57	92
Przesuwanie okna	400	148	301



KOPROCESOR Matematyczny

-%:=+

Wojciech [illegible]

Z [illegible] związane [illegible] na Amidze [illegible] komputer, [illegible] częściej niż jakikolwiek [illegible]ających formaty [illegible] skończywszy.

Artykuł ten szczególnie polecam osobom mającym koprocesor matematyczny w swojej Amidze (np. zainstalowany zgodnie z artykułem z Amigowca 2/94 ???) i chcącym poznać detale jego pracy. Nie chcę powtarzać treści artykułu o montażu FPU w A1200, podaję jednak kilka faktów:

- Amigi oparte są na procesorze z rodziny MC 68xxx.
- Koprocesor matematyczny (MC 68881/82 ang. Floating Point Unit) można bezpośrednio podłączyć do procesora MC 68020 i nowszych (Amigi 1200, 2500, 3000, 4000).
- Procesory MC 68008/10/12 (Amigi 500, 500+, 600, 1000, 1500 i 2000) nie pozwalają na bezpośrednią współpracę z koprocesorami. W tych komputerach koprocesor podłącza się jako urządzenie wejścia/wyjścia i dołącza programik zapewniający komunikację procesora i koprocesora.
- Procesory MC 68xxx pozwalają na podłączenie do 8 niezależnych koprocesorów, w tym 7 dowolnego typu (np. matematyczne). Koprocesor nr 0 musi być układem MMU (MC 68851, ang. Memory Management Unit - układ zarządzania pamięcią)
- Procesory MC 68040 i 68060 mają koprocesory: FPU, jak i MMU wbudowane wewnątrz układu. Podłączenie dodatkowych koprocesorów jest możliwe, lecz wymaga specjalnego oprogramowania i osprzętu.
- Koprocesor MC 68882 jest szybszą wersją koprocesora MC 68881.
- Połączenie między koprocesorem i procesorem głównym jest w pełni asynchroniczne. Oba układy mogą mieć różne zegary.

Instrukcje koprocesora FPU.

Procesor główny wykonuje program - sekwencję liczb, które są rozpoznawane przez niego jako instrukcje. Każda instrukcja jest równoważna jednej liczbie, jednak nie każda liczba jest ważną instrukcją. Pośród liczb oznaczających różne rozkazy procesora np. dodawanie, mnożenie itp. schowane są liczby nie rozpoznawane przez procesor. Są to instrukcje nielegalne - ich wykonanie zatrzyma aktualny program i wywoła odpowiedni program obsługi stanu wyjątkowego.

Firma Motorola zdefiniowała specjalną grupę instrukcji nielegalnych mających kod:

$Axxx lub $Fxxx

x - dowolna cyfra.

Obie grupy mają własne wektory przerwań, nazwane od pierwszych cyfr rozkazu jako emulatory linii F i A:

Linia A - wektor 10 - adres wektora $28

Linia F - wektor 11 - adres wektora $2C

Wprowadzając procesor MC 68020 i koprocesor MC 68881, Motorola opracowała specjalny protokół komunikacji z koprocesorem - używający instrukcji $Fxxx. Rozpoznanie tej instrukcji rozpoczyna cykle komunikacji z koprocesorem kończące się:

A. Błędem (nie ma koprocesora). Wywołany jest stan wyjątkowy linii F.

B. Przesłaniem do koprocesora odpowiedniego rozkazu i normalnym pobraniem kolejnej instrukcji.

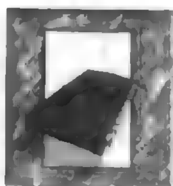
Rozkaz koprocesora zakodowany jest w samym słowie $Fxxx lub w kolejnym długim słowie/wielokrotności długiego słowa. Licznik rozkazów PC jest automatycznie modyfikowany, aby "przeskoczyć" ewentualne rozszerzenia do rozkazu $Fxxx. (Patrz Rysunek 1)

Instrukcje koprocesora mają dość dziwaczny format wewnętrzny, program Assembler automatycznie tłumaczy tradycyjne mnemoniki do kodu wynikowego. Mnemoniki rozkazów koprocesora mają format podobny do instrukcji procesora głównego np.:

FMOVE D0,FP1 ; Przesłanie liczby zmiennoprzecinkowej z rejestru D0 do rejestru koprocesora FP1.

Instrukcje koprocesora mają dostęp do wszystkich trybów adresowania procesora głównego, w tym do jego rejestrów. Pobieranie argumentów z pamięci wykonuje procesor główny, po dokonaniu wszelkich usług może kontynuować program główny wykonując instrukcje niezależnie od koprocesora. Jeśli procesor napotka kolejną instrukcję koprocesora, to w zależności od wstawionego układu:

68881 - Praca procesora głównego wstrzymywana jest do czasu, gdy zakończy się poprzednia instrukcja koprocesora. Odbiera wtedy jej wyniki, przesyła kolejny rozkaz koprocesorowi



i wykonuje jego żądania. Po spełnieniu wszystkich żądań wykonuje kolejny rozkaz.

68882 - Jeśli instrukcja koliduje z aktualnie wykonywaną przez koprocesor, procesor pobiera wszystkie potrzebne argumenty. Dalsze działania są identyczne z 68881. Gdy instrukcja nie koliduje ze statusem koprocesora, koprocesor otrzymuje kolejny rozkaz i wszystkie związane z nim argumenty. Procesor może kontynuować program.

Kolizja między instrukcjami 68882 to: praca kolejnych instrukcji na co najmniej dwóch identycznych rejestrach, zmiana/testowanie statusu procesora, dwie instrukcje arytmetyczne. Tak więc niemożliwe jest jednoczesne wykonywanie dwóch operacji arytmetycznych. Nie można również jednocześnie wykonywać operacji operującej na rejestrach używanych przez drugą instrukcję:

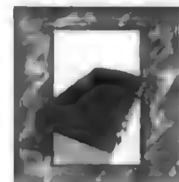
Instrukcja Argument -> FP5 ; Te instrukcje nie wykonają się jednocześnie.
Instrukcja FP5 -> Argument ; Czekam na zapełnienie rejestru FP5.

W najlepszym wypadku koprocesor MC 68882 może wykonywać instrukcję arytmetyczną (np. FMUL.x - mnożenie), mieć pobraną kolejną instrukcję arytmetyczną nie kolidującą z aktualnie wykonywaną oraz wykonywać instrukcję przesyłu danych (FMOVE) nie kolidującą z pozostałymi instrukcjami. Koprocesor MC 68881 całość wykonałby w czasie znacznie dłuższym niż suma czasów wykonania kolejnych instrukcji MC 68882 - przecież nie może pobierać argumentów podczas pracy nad poprzednią instrukcją. Dużo czasu zaoszczędzi procesor główny. Wymiana koprocesora MC 68881 na elektrycznie i programowo kompatybilny MC 68882 może przyspieszyć wykonywanie programu 2-5 razy.

Rejestry koprocesora i formaty danych.

Koprocesory MC 68881/82 mają zestaw rejestrów odrębny od rejestrów procesora głównego. Osiem rejestrów danych FP0-FP7 przechowuje dane biorące udział w obliczeniach. Rejestry te są w pełni uniwersalne, używając instrukcji koprocesora możemy operować na ich zawartości identycznie jak

Rysunek 1: Artykuł o użytkowanniu koprocesora. Schemat wymiany informacji między CPU i FPU. Przekazywanie rozkazu.





Typ (precyzja)	Mantysa rzeczywista
pojedyncza (ang. single)	Mr=1,<23 bity pseudomantysy>
podwójna (ang.double)	Mr=1,<52 bity pseudomantysy>
rozszerzona (ang.extended)	Mr= <64 bitowa mantysa>

Uwaga! Format rozszerzony jest zamieniany [redacted] 80bitowy format wewnętrzny (bez zer).

Stąd dotyczy on tylko przechowywania w pamięci operacyjnej.

Rysunek 2. Formaty danych zmiennoprzecinkoych.

robiliśmy to z rejestrami procesora głównego. Każdy ma długość [redacted] bitów.

Koprocesor pozwala na operowanie danymi o formatach:

- Stałoprzecinkowy: 8, 16 i 32 bity. Format uzupełnienia do dwóch, dane są identyczne jak obsługiwane przez procesor główny. Operacje przeprowadzane na tych danych to jedynie konwersje, np. pomiędzy rejestrami procesora i koprocesora.

- Zmiennoprzecinkowy: 32, 64 i 80 bitów. Dane składają się z mantysy i wykładnika. Podczas zapisu do pamięci 80 bitowa dane zamieniane są na 96 bitów.

- Dane BCD. To 24 cyfry dziesiętne, każda z nich zajmuje 4 bity z wartościami $A-F zakazanymi (format instrukcji SBCD i ABCD). Cała cyfra zapisywana jest na [redacted] bitach. (Patrz Rysunek 2)

Koprocesory Motoroli w praktyce nie liczą na wszystkich podanych typach danych, argumenty są zamieniane na wewnętrzny format 80 bitów. Jeśli instrukcja zapisuje wynik w innym formacie następuje druga konwersja. Konwersja następuje również przy przesyłaniu danych, np. z rejestru D0 do FP0:

FMOVE.L D0,FP0 ; 32 bitowa liczba z rejestru D0 jest przekształcana na 80 bitową liczbę zmiennoprzecinkową i zapisywana w FP0.

Koprocesor MC [redacted] posiada specjalny blok, który dokonuje konwersji niezależnie od jednostki arytmetycznej. Jest to kolejna cecha stawiająca go wyżej od 68881.

Koprocesor posiada również specjalne rejestry kontrolne, podobne do do rejestru <SR> z procesora głównego. Specjalne instrukcje skoków pozwalają na zmianę biegu programu w zależności od stanu znaczników koprocesora.

Biblioteki systemowe.

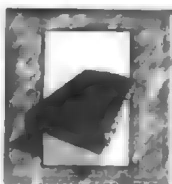
System operacyjny Amigi zastępuje dostęp do dowolnych zasobów sprzętowych za pomocą odpowiednich bibliotek. Są one zbiorem funkcji wykonujących określone działania - program użytkownika nie musi bezpośrednio odwoływać się do zasobów komputera tym samym zapewniając kompatybilność wszystkich modeli Amiga. Programy automatycznie korzystają z ulepszeń systemu operacyjnego i nowych urządzeń np. koprocesora matematycznego. Istnieje 6 bibliotek matematycznych:

MathFfp |
MathTrans | - Dane 32 FFP.
|

MathIeeeDoubBas |
MathIeeeDoubTrans | - Dane 64 bity Ieee.
|

MathIeeeSingBas |
MathIeeeSingTrans | - Dane [redacted] bity Ieee.
|

Ostatnie dwie biblioteki zostały wprowadzone dopiero z systemem 2.0, po



zostałe istniały od początku historii Amigi. Biblioteki automatycznie rozpoznają i korzystają, z koprocesora, gdy go nie ma korzystają z własnych programów emulujących jego obecność - jest to jednak znacznie wolniejsze.

Biblioteki zapewniają pracę na trzech formatach - 32 i 64 bitowe Ieee zgodne z formatem koprocesora oraz 32 bitowy FFP ułatwiający szybkie obliczenia. Format 80 bitowy nie jest chwilowo dostępny. Funkcje biblioteki zależą od fragmentu jej nazwy:

Bas (ang. BASic - podstawowe) - biblioteka zapewnia podstawowe funkcje matematyczne, np. dodawanie, odejmowanie, mnożenie.

Trans (ang. TRANScendental - trygonometryczne) - biblioteka zawiera wyższe funkcje matematyczne, np. Sin(x), Cos(x).

Wykorzystanie funkcji bibliotek matematycznych ma zarówno wady, jak i zalety. Do niewątpliwych zalet należą:

- Biblioteki pozwalają na wykorzystanie dowolnych "urządzeń" matematycznych, np. niestandardowego koprocesora.
- Korzystanie z bibliotek odciąża nas z problemu obsługi braku koprocesora i emulowania jego rozkazów.
- Nasz program będzie kompatybilny z przyszłymi procesorami/koprocesorami Motoroli, ktore mogą zachowywać sie odmiennie niż MC 68881/82.

Ich wykorzystanie ma również wady:

- Biblioteki te są "pamięciożerne" - do 100 KB podczas pracy.
- Bezpośrednie korzystanie z koprocesora jest szybsze.
- Biblioteki ograniczają możliwości koprocesora MC 68882.
- Algorytmy emulujące pracę koprocesora są wolniejsze niż bezpośrednie emulowanie stosowane w programach.

Jak korzystać z koprocesora? (Assembler).

Jeśli program nie musi być oszałamiająco szybki, najlepiej wykorzystać biblioteki systemowe Amigi. Nie trzeba będzie wykonywać dość niewygodnego dostępu do koprocesora, ponadto funkcje systemowe ustawiają znaczniki procesora głównego - nie będzie konieczności testowania znaczników koprocesora. Wykorzystanie funkcji systemowych może więc znacznie uprościć korzystanie z liczb rzeczywistych.

Sytuacja jest bardziej skomplikowana, gdy Twój program bardzo potrzebuje kolejnych godzin pracy komputera, warto wtedy stworzyć szybszą wersję bezpośrednio korzystającą z koprocesora. Komendy koprocesora będą bezpośrednio wpisane w kod programu, stąd nie ma potrzeby trzymania kodu emulacji koprocesora (pamięć!). Nie są potrzebne żadne instrukcje pomocnicze - program będzie wykonywany znacznie szybciej. Wreszcie mamy możliwość pełnego wykorzystania możliwości koprocesora MC 68882, czyli jednoczesnego wykonania kilku instrukcji. Programy wykorzystujące koprocesor w sposób "hardware'owy" powinny sprawdzać jego obecność i reagować ostrzeżeniem typu "Brak koprocesora" zamiast "wysypania" komputera i rameczki Guru. Warto dodać wersję tylko na komputery bez koprocesora, np. stosując własne typy danych.

Możemy wreszcie wykorzystywać instrukcje koprocesora odpowiednio zabezpieczając wektory przerwań odpowiedzialne za ten układzik. Jednak w sytuacji braku koprocesora programu jest jeszcze wolniejszy w działaniu niż z wykorzystaniem bibliotek Math.

Uwaga!

Bibliotek matematycznych nie wolno wykorzystywać w procedurach obsługi przerwań, gdyż w przypadku braku koprocesora przydzielają one dodatkową pamięć. Przydzielanie pamięci podczas przerwań często doprowadza do pewnego czerwonego prostokąta...

Jak korzystać z koprocesora? (Języki kompilowane).

Języki te przeważnie korzystają z formatu liczb rzeczywistych (ang. Real) podczas dowolnych operacji. Stąd możemy bezpośrednio operować wszystkimi dostępnymi instrukcjami, zmieniając jedynie format danych - na przykład podczas deklaracji. Wszelkie operacje związane z użyciem koprocesora bedą automatycznie obsługiwane przez kompilator. Większość kompilatorów pozwala jednocześnie określić sposób generacji kodu:

- Bezpośrednie użycie koprocesora.
- Wykorzystanie bibliotek Math.

Zwróć uwagę, że faktyczna kontrola nad sprzętem jest mniejsza niż wykorzystanie bibliotek Math. Jeśli bardzo zależy Ci na prędkości, proponuję użycie tych bibliotek. Najszybszym rozwiązaniem będzie oczywiście użycie Assemblera i instrukcji koprocesora.

Inne jezyki. (Interpretery)

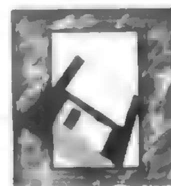
Chodzi mi o języki "dla leniwych" czyli AMOS, różne odmiany Basica, nietypowe interpretery oraz niektóre kompilatory. Są one nastawione na jak najszybsze tworzenie programu bez potrzeby wnikania w mechanizmy niskiego poziomu (Assembler czy nawet typy danych) oraz system operacyjny, a głównie w biblioteki tworzące środowisko i interface użytkownika (Intuition.library, Graphic.library, itd.). Takie traktowanie ułatwia błyskawiczne pisanie programów uniwersalnych, jednak zastosowanie tych języków do prac specjalizowanych, np. tworzenia oryginalnego interface'u jest bardzo utrudnione, jeśli nie niemożliwe.

Ta sama sytuacja występuje przy próbach wykorzystania specjalizowanych typów danych. Środowiska te zapewniają formaty liczb zmienno i stałoprzecinkowych (czasem tylko jeden z typów), jednak bez możliwości ingerencji w detale tych danych. Wykorzystanie koprocesora jest bardzo nieefektywne, co gorsze nie ma możliwości na łatwą poprawę sytuacji.

Jedną z możliwości jest wykorzystanie bloków kodu generowanych przez inny język np. C czy Assembler. Fragmenty wykorzystujące koprocesor lub emulujące jego pracę integrujemy z programem interpretera, uzyskując w miarę szybką pracę programu.

W lepszych interpreterach mamy możliwość wywoływania funkcji systemowych, w tym bibliotek matematycznych. Jeśli dodatkowo możemy przechowywać ich rezultaty (przynajmiej wskaźniki), to można ważniejsze obliczenia wykonywać z ich wykorzystaniem.

Jak widać problematyka koprocesora matematycznego nie jest zbyt łatwa. Mam nadzieję, że choć w jakimś stopniu przybliżyłem Wam zasady działania i wykorzystanie tego bardzo ważnego elementu komputera. ❑



REAL 3D v.1.4

Sebastian Dingski

REAL 3D jest rewelacyjnym programem do RAY TRACINGU. Porównywać można go właściwie do tylko kilku innych programów takich jak IMAGINE czy 3D PRO. Program ma bardzo wiele zalet natomiast bardzo mało wad. Właśnie to stawia go w czołówce programów tego typu. W dziale kreowania Reala 3D v1.4 postaram się przybliżyć starszą, lecz dla niektórych nieznaną jego wersję.

Edytor REALa jest podzielony na trzy zasadnicze części:
Edytor główny - edytor, w którym definiujemy obiekty oraz inne elementy składające się na nasz projekt.
Wireframe - edytor umożliwiający ustawienie punktu, z którego widziany jest dany obiekt. Pomocny także w robieniu prostych animacji.
Solid - ekran na którym możemy ustawić parametry dotyczące generacji obrazka.

Edytor główny REALa podzielony jest na cztery okienka. Trzy z tych okienek przedstawiają rzuty figury z różnych punktów, tzn. z przodu, z boku i z góry. W czwartym okienku znajduje się około 50 ikonek służących do wybierania najczęściej używanych opcji takich jak: postawienie światła, stworzenie podstawowych obiektów, czy też ustawienie obserwatora czyli punktu z którego oglądamy nasze dzieło. Wszystkie opcje możemy także wybrać z menu, które jest podzielone na:

PROJECTS - w menu tym zawarte są wszelkie operacje dotyczące wczytywania i nagrywania obiektów, materiałów, czy też gotowych animacji. Parametry ekranu edytora także ustawiamy w tym menu. Możemy stworzyć dowolny materiał lub macro-funcję, która przydaje się przy bardziej skomplikowanych figurach. PROJECTS ma także dość obszerne podmenu służące do tworzenia animacji. Jest ono główną siłą do jej robienia w tym programie. Dzięki niemu możemy kasować, dodawać, "zamrażać" poszczególne klatki, a także wykonywać dużo bardziej skomplikowane operacje jak chociażby MORPHNING, czyli płynną zamianę jednego obiektu w drugi.

CREATION - menu to służy do definiowania brył. Możemy także ustawić źródło światła oraz punkt, z którego patrzymy. CREATION zawiera dwa podmenu: PRIMITIVES i TOOLS. To pierwsze pozwala nam na stworzenie zdefiniowanych przez autorów programu standardowych figur, takich jak np. sześcian, stożek, trójkąt. Drugie zaś pomaga nam przy tworzeniu trudniejszych obiektów. Dzięki TOOLS możemy łatwo, szybko i przyjemnie zrobić, np. kielich.

FREEFORM - zawarte w nim narzędzia pozwalają na tworzenie obiektów z "wolnej ręki", a także na rozmaite zniekształcenia zdefiniowanych figur.

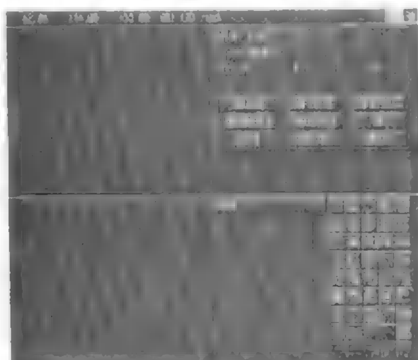
MODIFY - jest to olbrzymie menu pozwalające na rozmaite modyfikacje zrobionych przez nas brył. Możemy przesunąć, rozciągnąć, powielić, skasować, obrócić dowolny obiekt. Tutaj nadajemy figurze materiał lub kolor.

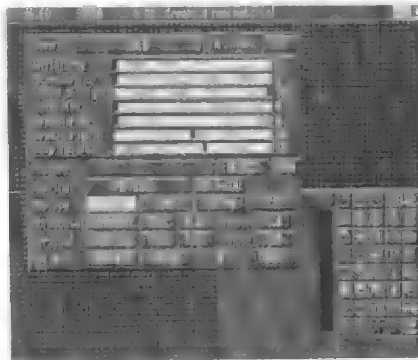
COLORS - służy do zmiany kolorów, i do ustawienia palety kolorów ekranu edytora.

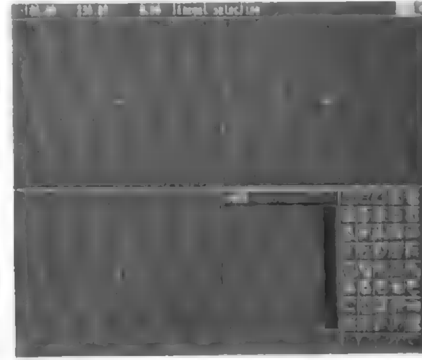
SETTINGS - zawarte w nim opcje dotyczą sposobu rysowania figur na ekranie edytora.

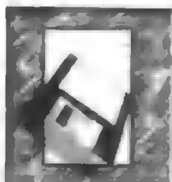
EXTRAS - zawiera kilka pomocniczych funkcji, jak np. podawanie ilości wolnej pamięci, podawanie informacji o danych obiektach, czy też funkcja UNDO.

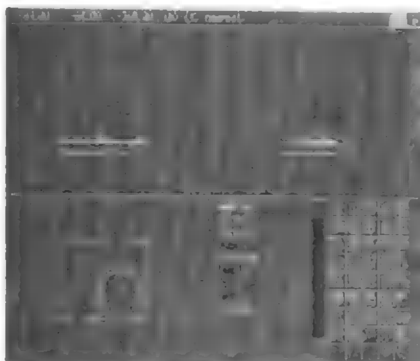
MODES - Służy do przechodzenia do poszczególnych części REALa.











Kolejną częścią programu jest **Wireframe**. Służy ona do ustawiania punktu obserwatora. Większą powierzchnię ekranu zajmuje wektorowy obraz stworzonej przez nas sceny. W dolnej części ekranu znajdują się ikonki służące do przechodzenia pomiędzy klatkami, ustawiania odległości, czy też nagrywania animacji. Dość ciekawą opcją jest RBOX, która zamienia skomplikowane bryły na prostopadłościenne klocki, dzięki czemu rysowanie jest o wiele szybsze.

Ostatnią częścią programu jest **Solid**. Możemy tutaj ustawić sposób generowania grafiki. Tak więc mamy do wyboru:

FAST - tryb, w którym komputer rysuje najszybciej. Nie uwzględnia jednak materiałów i cieni.

LAMPLESS - tryb, w którym zostaje odjęta pewna ilość światła.

SHADOWLESS - tryb, w którym zmniejszona jest ilość cieni.

OUTLINE - obrysowywanie danych obiektów.

NORMAL - pełny RAY TRACING z uwzględnieniem świateł, cieni i materiałów.

W tej części możemy ustawić rozdzielczość, wielkość punktów (wysokość oraz szerokość pojedynczego punktu - 1,2,4,8), a także tryb graficzny - HAM, INTERLACE, 24 bity, TARGA, czy też obrazek w odcieniach szarości. Obrazek taki jest generowany w rozdzielczości 640*512 przy użyciu 16 odcieni szarości. Jego jakość jest rewelacyjna, z trudem można odróżnić go od fotografii. W trybie tym REALowi nie dorównuje żaden inny znany mi program!

Poniżej prezentujemy szczegółowy opis opcji Reala 1.4 , gdyż jest to wersja niedostępna na rynku już od paru ładnych miesięcy.

Menu Projects

Objects:

Create - Tworzy w wybranym zbiorze obiektów kolejny zbiór o nazwie OBJ.

Create Root - Tworzy nowy zbiór główny o nazwie ROOT.

Load - Wczytywanie obiektów z dysku.

Save - Nagranie wybranego obiektu.

Animation:

Size - Określa ilość klatek animacji. Po zwiększeniu ilości klatek każda otrzymana tym sposobem klatka wygląda jak klatka źródłowa (zawiera te same obiekty oraz ich ułożenie). Przy ilości klatek większej niż 1, na górnej listwie programu pojawiają się strzałki służące do poruszania się pomiędzy poszczególnymi klatkami oraz znak (X). Znak ten służy do "zamrażania" danych klatek.

Delete - Kasowanie całej animacji, także klatki głównej.

Insert - Dodawanie określonej ilości klatek do tworzonej animacji.

Remove - Kasowanie wybranej ilości klatek łącznie z właśnie edytowaną.

De-expose - "Odmrażanie" klatek.

Load - Wczytywanie animacji (pojedynczej klatki jako projekt lub też zbioru klatek). Po wczytaniu animacji pojawia się okienko z zapytaniem czy wgrana animacja ma zostać połączona z aktualnie edytowaną (Join animations), czy też wgrana jako kolejna klatka (Concatenate animations). Istnieje także inna możliwość wgrania animacji. Opcja ta (Replace old animations) wgrywa nową animację na miejsce aktualnie tworzonej powodując tym samym jej skasowanie.

Save - Zapisuje edytowaną animację na dysk. Zapisane zostają wszystkie obiekty składające się na daną animację wraz z ich ułożeniem, kolorami, materiałami, itp.

Orbit - Tworzy ścieżkę ruchu dla zaznaczonego obiektu. Po wybraniu tej opcji rysujemy w jednym z okien drogę dla danego obiektu. Po naciśnięciu prawego przycisku program zapyta się o zakres klatek w jakich ma być użyta zdefiniowana droga.

Rotation - Obraca zaznaczony obiekt wokół określonego punktu. Obrót określamy w stopniach.

Direction - Opcja ta powoduje, że kierunek ruchu obiektu ze zdefiniowaną wcześniej ścieżką ruchu (opcja Orbit) zawsze będzie względem kierunku ruchu.

Key framing - Przekształcenie liniowe dwóch nie różniących się ilością punktów obiektów. Przekształcenie możemy uzyskać dla obiektów o tej samej nazwie pomiędzy klatkami "zamrożonymi".

Morphing - Opcja podobna do poprzedniej z tym, że przekształcenie nie jest liniowe, lecz polega na przesunięciu punktów danej figury.

Goto frame - Przeskoczenie do klatki o podanym numerze.

Play - Odtworzenie animacji w oknach edytora.

Materials:

Create - Tworzenie nowych materiałów.

Modify - Modyfikacja stworzonego uprzednio materiału.

Delete - Kasowanie materiałów. Opcja ta skasuje wszystkie materiały (Remove all materials) lub jeden wybrany (Remove one material).

Load - Wgrywanie materiałów z dyskietki. Program może wgrać materiały kasując już istniejące lub dołączyć je do już używanych.

Save - Nagrywanie materiałów na dyskietkę.

Macro:

Define - Tworzenie makrodefinicji, tzn. zbioru poleceń jakie ma wykonać program przy każdym wywołaniu makrodefinicji. Bardzo pomocne przy tworzeniu projektów z wielu identycznych obiektów.

End - Zakończenie makrodefinicji.

Execute - Wywołanie makrodefinicji.

Screen:

(Pokazuje się okienko, w którym możemy wybrać tryb ekranu, w którym pracujemy.)

Interlace - Włączenie lub wyłączenie trybu INTERLACE.

NTSC - Tryb wyświetlania ekranu o ilości linii równej 200 lub 400 (z włączonym INTERLACE).

Screen depth - Ilość bitplanów używanych przez program. Im mniejsza ilość tym więcej pamięci, ale mniej kolorów: 2 bitplany - 4 kolory, 3 bitplany - 8 kolorów, 4 bitplany - 16 kolorów.

Aspect ratio - Stosunek wysokości do szerokości ekranu. Standardowo 1,0.

Overlap - Włączenie tego parametru spowoduje, że aktywne okno edytora będzie miało wymiary prawie całego ekranu.

Exit:

Wyjście z programu.

Menu Creation

Primitives:

(Tworzenie obiektów)

Triangle - Trójkąt.

Rectangle - Prostokąt.

Circle - Koło.

Prism - Pryzmatoid.

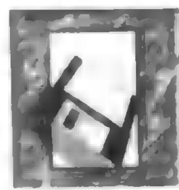
Cube - Sześcian, a w zasadzie prostopadłościan.

Pyramid - Ostrosłup o podstawie prostokąta.

Sphere - Kula.

Cone - Stożek.

Cylinder - Walec.



Hyperbol - Wklęsły walec.
Cut hyperbol - Ścięty stożek.

Tools:

(Tworzenie brył z obiektów typu primitives.)

Lathe - Tworzenie bryły obrotowej. Po wybraniu opcji należy narysować oś bryły, a następnie przekrój obiektu. Po zakończeniu rysowania przekroju zostanie on obrócony wokół osi tworząc w ten sposób bryłę obrotową. Narzędziem tym możemy w bardzo łatwy sposób stworzyć w wszelkie wazony, kielichy, itp.

Circular tube - Tworzy rurę o przekroju okrągłym. Przekrój rury, a tym samym jej średnicę definiuje koło.

Rectangular tube - Tworzy rurę o przekroju prostokątnym.

Fence - Tworzy równoległościan o zdefiniowanej płaszczyźnie. Przydat■■ przy tworzeniu prostych liter, np takich jak L, I, E, T.

Pixel tool - Zamienia obrazek w formacie IFF na obiekt. Wszystkie pixele obrazka o kolorze innym od tła zostaną zamienione na wybrany obiekt. Przy czym kolor danego elementu zależy od koloru danego pixela.

Polygon - Tworzenie płaszczyzny o dowolnym kształcie i dowolnej ilości krawędzi.

Polyhendron - Tworzenie bryły o dwóch równoległych ścianach. Z tym, że podstawy tej figury mogą mieć różne wielkości.

Conical tube - Rura okrągła o zmiennej średnicy. Po jej narysowaniu określamy ile obiektów ma mieć każdy z narysowanego węzła (liczba ta określa dokładność figury - im więcej tym lepsza jakość, ale także znaczne spowolnienie pracy oraz większa zajętość pamięci).

Lathe 2 - Opcja podobna do Lathe z tym, że stworzone przez nią obiekty wydłużają generowanie obrazka. Wpływa to na jego jakość - powierzchnie są bardziej okrągłe.

Pixel tool 2 - Tworzy obiekt typu siatkowego (mesh) na podstawie wgranego obrazka w formacie IFF. Im jaśnieszy kolor czerwony pixela na obrazku, tym większe pofałdowanie naszego obiektu.

Lamp:

Ustawianie światła. Przy ustawianiu światła należy zwracać uwagę wyłącznie na kolor, który wpływa ■■ jego jasność. Wielkość oraz kierunek nie wpływają na parametry światła.

Observer:

Ustawienie punktu, z którego oglądamy nasz projekt.

Aim point:

Ustawienie punktu, na który patrzymy.

Menu Freeform

Create curve:

Curve - Tworzenie krzywej łamanej. Przy rysowaniu krzywej użycie klawisza ESC powoduje skasowanie ostatnio narysowanego punktu.

Circular loop - Tworzenie zamkniętej krzywej w kształcie okręgu o podanej dokładności (im większy parametr, tym bardziej okrągła krzywa).

Spiral - Pomaga w robieniu spirali, tzn. tworzy krzywą spiralną. Zaznaczamy wielkość spirali, ■ następnie podajemy kąt obrotu, liczbę podpodziałów (oznaczającą stopień złagodzenia ostrych rogów) oraz długość.

Parallel - Rysowanie nowej, podobnej do zaznaczonej krzywej w podanej doległości.

Modify:

Show spline - Pokazuje jak będzie wyglądała krzywa po zaokrągleniu.

Remap - Zmiana ilości punktów krzywej lub obiektu siatkowego. Przy zmianie na większą ilość punktów krzywa zostanie zaokrąglona.

Smoothen - Zaokrąglanie krzywej. Nie zmienia ilości punktów.

Close - Zamykanie krzywej. Polega na połączeniu pierwszego i ostatniego punktu krzywej.

Break - Działa przeciwnie niż poprzednia funkcja, tzn. rozwiera zamkniętą krzywą.

Concatenate - Łączy dwie krzywe w jedną. Po wybraniu opcji należy pokazać krzywą, z którą ma połączyć się krzywa wcześniej zaznaczona oraz należy zaznaczyć końce krzywych, którymi mają się złączyć.

Remove points - Kasowanie zaznaczonych punktów z krzywej (bez jej przerywania)

Subdivide - Dodaje określoną ilość podpodziałów do krzywej lub obiektu siatkowego zwiększając tym samym jej dokładność.

Build:

Coplanar sweep - Tworzenie obiektu siatkowego o przekrojach równoległych.

Orthogonal sweep - Tworzenie obiektu siatkowego o przekrojach utrzymujących stały kąt względem kierunku tworzącej siatkę.

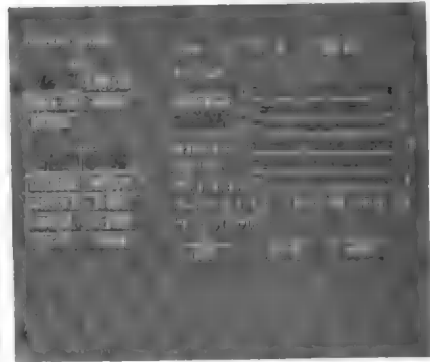
Rotation - Tworzy bryłę obrotową z siatki.

Swing - Tworzy bryłę obrotową z tym, że oprócz linii brzegowej i osi obrotu musimy podać przekrój z rzutu poziomego.

Join - Łączy dwie krzywe tworząc obiekt siatkowy.

Bending modes:

(Ustawienie sposobu wyginania obiektów.)



Bend and Move - Zwrot wyginania nie zależy od położenia interwału (interval).

Bend and Size - Zwrot wyginania będzie przeciwny po różnych stronach interwału.

2D - Obiekt będzie wyginany niezależnie od położenia interwału w innych rzutach.

3D - Obiekt będzie wyginany w zależności od położenia interwału w innych rzutach.

Radial - Wyginanie odbędzie się w kierunkach promienistych.

Bend:

Local - Wyginanie całego obiektu w zależności od długości interwału.

Global - Wyginanie całego obiektu niezależnie od wielkości interwału.

End point - Wyciąganie obiektu. Punkt nieruchomy znajduje się tam gdzie początek interwału.

Linear - Wyginanie liniowe (bez efektów zagięć, itp.)

Point editing:

Select - Zaznaczenie nowych punktów.

Select new - Zaznaczenie nowych punktów wraz ze skasowaniem (zaznaczenia) starych.

Deselect - Skasowanie zaznaczenia wybranych punktów.

Deselect all - Skasowanie zaznaczenia wszystkich punktów.

Show point - Pokazanie wszystkich punktów wybranego obiektu.

Menu Modify

Hierarchy:

Move - Przesuwanie obiektu.

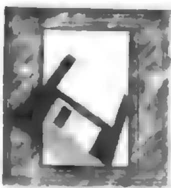
Move to - Przesunięcie obiektu do wyznaczonego miejsca.

Strech - Zmiana wielkości figury w aktualnie zaznaczonym oknie, tzn. tylko względem dwóch osi, np. szerokość i wysokość.

Size - Zmiana wielkości obiektu we wszystkich kierunkach.

Rotate - Obracanie obiektów o dowolny kąt.

Mirror - Odwrócenie zaznaczonego obiektu względem narysowanej osi.



Extend - Rozciąganie obiektu w dowolnych kierunkach.

Explode - Rozciąganie zaznaczonego zbioru obiektów bez zmiany ich kształtu oraz wielkości.

Copy - Kopiowanie zaznaczonej figury.

Rename - Zmiana nazwy figury lub też katalogu (zbioru obiektów).

Locate - Przeniesienie obiektu lub zbioru obiektów do innego zbioru.

Delete - Skasowanie obiektu lub katalogu.

Color - Zmiana koloru.

Material - Zmiana materiału dla danego obiektu.

Painting - Ustawienie miejsca nałożenia textury.

Wildcard:

Color - Zmiana koloru dla wszystkich obiektów o podanej nazwie, kolorze lub materiale.

Replace - Zmiana obiektu o podanych parametrach (podobnie jak wyżej) na wybrany.

Delete - Kasowanie obiektów o podanych parametrach.

Macro - Zastosowanie zdefiniowanej wcześniej makrodefinicji do obiektów o podanych atrybutach.

Operations:

AND - Iloczyn logiczny. Zostawia tylko część wspólną dwóch obiektów.

EOR - Różnica symetryczna. Zostawia wszystko poza częścią wspólną obiektów.

AND NOT - Iloczyn z negacją. Wycina część wspólną oraz obiekt, którym dokonywaliśmy wycięcia.

AND with paint - Funkcja podobna do AND z tym, że zostaną zachowne parametry obiektów.

AND NOT with paint - Podobnie jak funkcja poprzednia tylko w odniesieniu do AND NOT.

DIVIDE - Dzieli podany obiekt na dwie części. Należy również podać obiekt dzielący.

Menu Colors

Wybór koloru oraz zmiana palety.

Menu Settings

Display:

Scale in - Powiększenie całej sceny w celu dokladniejszych poprawek.

Scale out - Pomniejszenie całej sceny.

Pan - Przesunięcie projektu tak, aby zaznaczone pole znalazło się w centrum okna.

Autofocus - Zmiana pola widzenia w ten sposób, aby cały zaznaczony obiekt znalazł się w centrum okien.

Reset - Ustawienie standardowego pola widzenia.

Drawmode:

Normal - Po każdej czynności związanej z obiektami wszystkie okna zostaną odświeżone.

Reduced - Odświeżane będzie tylko aktywne okno.

None - Żadne z okien nie będzie odświeżane.

Drawlevel:

All - Przy odświeżaniu zostają narysowane wszystkie obiekty.

Parent - Przy odświeżaniu zostaje narysowany obiekt zaznaczony wraz z innymi obiektami w tym samym katalogu.

Current - Narysowany zostanie tylko wybrany obiekt.

Coordinates:

Absolute - Koordynaty względem punktu zerowego.

Relative - Koordynaty względem pozycji kursora.

Abs and Rel - Automatyczne przełączanie pomiędzy Absolute, a Relative.

None - Wyłączenie współrzędnych.

Attributes:

Ustawianie nowych atrybutów dla obiektów typu PRIMITIVES. W menu tym możemy zmienić następujące rzeczy:

Name - Nazwa. Jeżeli nam nie odpowiada, np. TRIANGLE możemy zamienić nazwę na TRÓJKĄT i od tego czasu każdy z nowo stworzonych trójkątów będzie miał właśnie taką nazwę.

Flags - Atrybuty ustawiane w taki sam sposób jak w menu INFO.

Depth - Ustawianie wysokości dla brył, np. sześcian.

Custom name - Przy tworzeniu każdej figury program zapyta się o jej nazwę.

Custom depth - Przy tworzeniu każdej bryły program zapyta się o jej wysokość.

Save - Zgrywanie preferencji na dysk.

Alignment:

Przesunięcie dla skopiowanych obiektów.

Grid:

Ustawianie parametrów siatki. Określamy tu rozmiar siatki oraz to, czy ma ona być widoczna, czy nie.

Menu Extras

Redraw:

Odświeżenie ekranu.

Info:

Informacje o danym obiekcie. Ponadto możemy zmienić niektóre parametry objektu:

Color - Zmiana koloru.

Hollow - Obiekt nie będzie załamywał światła.

Uncovered - Dany obiekt będzie odkryty.

Scene - Obiekt ten nie zostanie narysowany podczas renderingu (procesie tworzenia obrazka). Przyspiesza liczenie obrazka.

Infinite - Obiekt nieskończenie wielki. Przydaje się przy tworzeniu podłoża lub nieba.

Invisible - Obiekt pozostanie niewidoczny podczas edycji. Zwiększa prędkość odświeżania okien.

Smooth - Wygładzanie brzegów w obiektach typu siatkowego.

Noreflect - Obiekt nie jest odbijany w innych figurach (przyspiesza proces renderingu).

Costs:

Set price - Ustawianie ceny obiektu.

Look price - Oglądanie ceny obiektu.

Avail mem:

Pokazuje ilość wolnej pamięci z podziałem na CHIP i FAST.

Get memory:

Ustawia preferencje REALa w ten sposób, aby zajmował on jak najmniej miejsca w pamięci, tzn. zostaje włączony tryb NTSC, cztery kolory, itp.

Representation:

Add wire - Wybrany obiekt będzie reprezentował katalog.

Delete wire - Skasowanie obiektu, który reprezentuje dany katalog.

Obscure - Skasowanie pojedynczego punktu reprezentacji.

Draw wire - Rysowanie reprezentacji.

Rethink - Automatyczna zmiana reprezentacji obiektów, na których zostały użyte funkcje typu AND, itp. Opcja ta pokazuje rzeczywisty wygląd obiektu.

No Wbench:

Zamyka okna Workbencha.

No icons:

Nie nagrywa ikon do projektów, obrazków, itp.

Undo on:

Włączenie funkcji undo.

Undo:

Przywraca stan projektu sprzed wywołania ostatniej opcji.

Menu Modes

Wireframe:

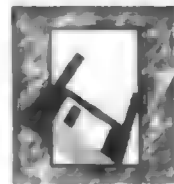
Przejście do edytora WIREFRAME.

Solid:

Przejście do edytora renderingu.

To już wszystko na dziś. W jedym z kolejnych numerów Amigowca postaram się bazując na informacjach zawartych w niniejszym artykule, zapoznać Was z techniką tworzenia scen, animacji, materiałów, itd.

REAL 3D v1.4 jak widać na załączonych ekranach jest programem wykonanym niezwykle estetycznie, a obrazki stworzone na tym programie (strony 75 i 76), co także jest widoczne na zdjęciach, mogą zaskoczyć niejednego. Jest to program, który ze względu na swoje niezbyt wielkie wymagania sprzętowe może być przydatny dla szerokiego grona miłośników grafiki 3D. ❑





MAXON CINEMA

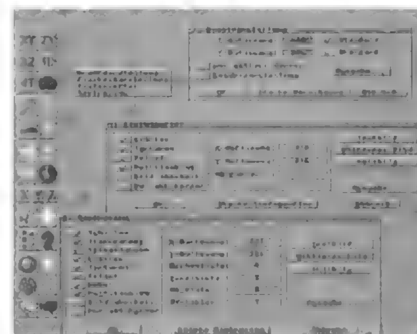
Łukasz Łukoda

Gdyby wszystkie firmy softwarowe zajmujące się tworzeniem oprogramowania na Amigę poszły za przykładem firmy Maxon, wówczas jedynym problemem użytkowników tych mikrokomputerów byłaby klęska urodzaju. Tak się jednak nie dzieje, dlatego tym bardziej należy życzyć Maxonowi dalszego, równie burzliwego rozwoju.

Po raz pierwszy firma Maxon zwróciła na siebie uwagę w roku 1991, za sprawą błyskotliwego programu "Maxon Cad 1.1". Od momentu pojawienia się, aż do chwili obecnej, jest to najszybszy ze znanych mi programów do wspomagania rysunku technicznego. Wkrótce później ukazała się wersja 2.0. Z żalem pomyślałem wówczas, że oto kolejny obiecujący producent oprogramowania ograniczy się tylko do jednego rejonu softwarowego rynku, wypuszczając kolejne wersje swojego sztandarowego programu. Na szczęście dla nas wszystkich, srodze się pomyliłem. Po krótkiej przerwie, kiedy wydawało się, że firma zapadła w letarg nastąpił niespodziewany "boom". "Maxon Fonts", "Maxon ASM", "Maxon C++", "Maxon PLP 2", "Maxon Viruscope 2", "Maxon Word", "Maxon Hothelp", "Maxon Magic", "Maxon Edward", "Maxon Sigmath 2", "Maxon Paint", "Maxon Twist", "Maxon Cinema 4D"... Uff! Takim repertuarem niewielu może się poszczycić. A to jeszcze nie wszystko. Istnieje bowiem, cały szereg różnego typu rozszerzeń do istniejących już programów, w postaci dysków z: fontami, symbolami i normami (do "Maxon Cada"), obiektami (do "Maxon Cinema 4D") itd. Wydaje się, że ambicją firmy stała się chęć posiadania w swojej ofercie wszystkich możliwych rodzajów oprogramowania. Do realizacji tego zadania niewiele już pozostało zważywszy na to, co już istnieje (DTP, baza danych, CAD, edukacja, program antywirusowy, program wspomagający elektroników, program graficzny, języki programowania...). Wszystkie produkty Maxona charakteryzują się pewnymi cechami wspólnymi. Do najważniejszych z nich należą: całkowicie bezkolizyjna praca z każdą wersją systemu (od 1.3 do 3.0), niska cena produktu (w porównaniu do otrzymywanych możliwości) oraz unifikacja interface'u użytkownika (większość programów firmy pracuje w systemie oddzielnych okien, o charakterystycznym ułożeniu znajdujących się w nich funkcji).

Jednym z najmłodszych produktów wydanych przez Maxona, jest tytułowy "Maxon Cinema 4D". Należy on do grona programów potrafiących tworzyć i animować obrazy metodą raytracingu. Jego twórcy wyszli z założenia, że wchodząc na arenę tego typu oprogramowania, nie mają szans na rywalizację z takimi kolosami jak "Imagine" czy "Real". W rezultacie tego stworzono program przeznaczony dla "mniej zamożnych" użytkowników, zachowujący przy tym szereg elementów typowych dla bardziej "elitarnych" produktów.

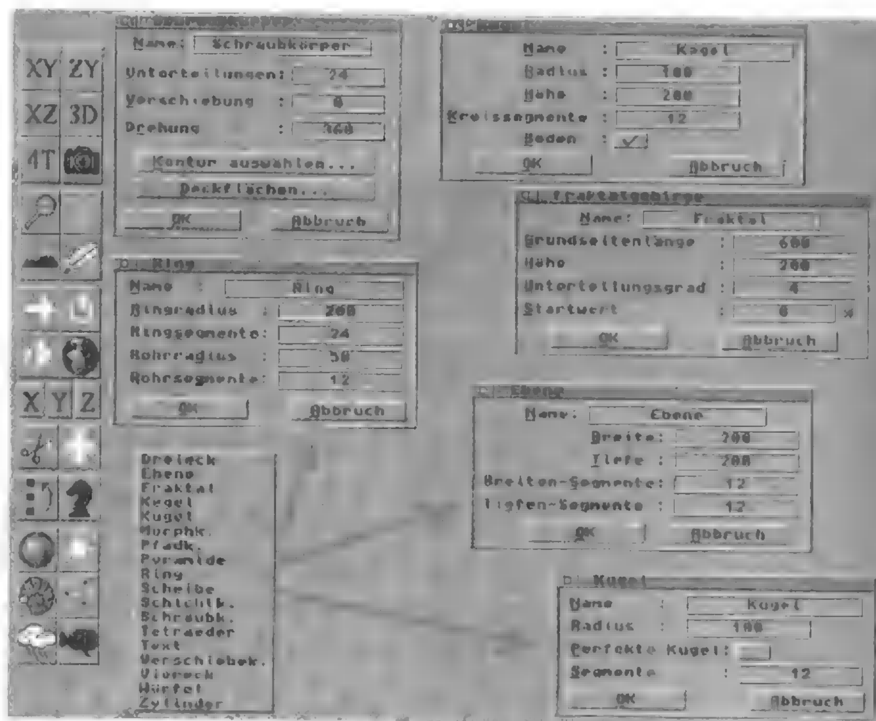
Podstawową zaletą "Maxon Cinema 4D" jest jego niezwykle niska cena (ok. 270 DM, dla porównania program "Aladdin 4D" kosztuje ok. [redacted] DM) pozwalająca, aby oryginał programu mógł trafić "pod strzechy" przeciętnych fanatyków komputerowej grafiki. Drugą niezwykle ważną cechą jest możliwość pracy na dowolnym modelu Amigi posiadającym 1,5 MB RAM. Program wraz ze wszystkimi dodatkami mieści się na siedmiu dyskietkach. Na pierwszej z nich znajdują się: główny program oraz specjalny dodatek w postaci modułu zatytułowanego "Raytracer". Ten ostatni powinien szczególnie ucieszyć posiadaczy minimalnej, wymaganej ilości RAMu. Jest to bowiem programik pozwalający na generowanie obrazków stworzonych w edytorze programu głównego, bez konieczności jego uruchamiania. Na pozostałych dyskach można znaleźć: program do konwersji obiektów (rozpoznaje on standardy: "Imagine", "Sculpt", "Turbo Silver" - wcześniejsza wersja "Imagina", "Fastray", "Reflections", "Videoscape", "Pagerender" oraz DXF - pochodzący od "AutoCADA"), przykładowe obrazki i sceny, tekstury, atrybuty (szkło, plastyk, metal, itp.), literki 3D, programiki do tworzenia i łączenia ze sobą animacji, drivery do popularnych kart graficznych ("Retina", "EGS") itd. Bardzo ciekawym pomysłem było dołączenie niewielkiego programiku ("Wolken") potrafiącego generować obrazek przedstawiający widok nieba wraz z chmurami. Tego typu "widoczek" jest częstym elementem towarzyszącym różnego typu scenom, stąd też jego przydatność jest oczywista. Tyle tytułem wstępu, przyjrzyjmy się teraz bliżej głównemu programowi.



Animation menu.

Po uruchomieniu "Maxon Cinema" wita nas niezwykle elegancko dopracowanym ekranem roboczym. Oczywiście jego centralną część zajmują okna rzutni: "X-Z", "Y-Z", "X-Y" oraz "Perspective". Na ekranie mogą znajdować się wszystkie cztery okna równocześnie lub jedno odpowiednio powiększone. Umieszczone z boku suwaki pozwalają na płynne skalowanie i przesuwanie całych edytowanych scen (dotyczy to również widoku w oknie "Perspective", w którym jest przedstawione położenie obiektów, w sposób jaki "widzi" je kamera). Poszczególne widoki zostały zaopatrzone w symbole odpowiadających im osi prostokątnego układu współrzędnych oraz w pomocnicze "siatki" ułatwiające precyzyjne ustawianie obiektów. Z prawej strony ekranu znajduje się zespół "przycisków" umożliwiających łatwy i bezpośredni dostęp do większości opcji tradycyjnego, rozwijanego menu.

Na szczególną uwagę zasługuje praca z obiektami, ponieważ element ten stanowi najmocniejszą stronę programu. Wspomniany wcześniej edytor jest nie tylko bardzo szybki, ale również starannie przemyślany. Wszystkie elementy, składające się na naszą scenę, mogą być dowolnie skalowane, obracane i przemieszczane nie tylko w podstawowych oknach układu rzutni, ale również bezpośrednio w widoku "Perspektive". W połączeniu ze wspomnianą wcześniej możliwością skalowania i przemieszczania całości, pozwala to na łatwe ustawienie najdogodniejszego "widoku" z kamery. Wskazanie interesującego nas obiektu odbywa się poprzez otoczenie go prostokątną ramką (w ten sposób mogą być zaznaczane całe grupy obiektów) lub wskazanie dowolnego punktu do niego należącego. Również samo tworzenie obiektów nie nastręcza zbyt wielu problemów. Obok szeregu "wbudowanych" na stałe elementów (stożki, kule, walce, płaszczyzny, torusy, ostrosłupy, a nawet coś przypominającego kształtem góry!) istnieje cały zestaw udogodnień, pozwalających na łatwe uzyskanie dowolnego kształtu. Niezwykle pomocny jest przy tym specjalny "edytor krzywych" pozwalający na łatwe otrzymanie nawet najbardziej skomplikowanych zarysów. Z nich to w dalszej kolejności uzyskuje się bryły, stosując takie opcje jak tłoczenie lub obrót dookoła osi. Istnieje również możliwość bezpośredniego przemieszczania poszczególnych punktów w obiekcie.



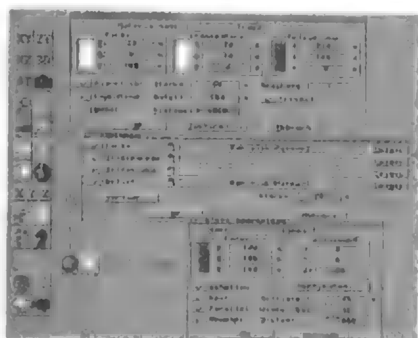
To tylko niektóre ■ opcji związanych z tworzeniem obiektów.

Odpowiednim elementom nadaje się atrybuty lub tekstury. W przypadku atrybutów określa się kolor przedmiotu, stopień przezroczystości, ilość światła odbitego, połysk i odblaski. Tekstur natomiast można używać jako map: kolorów obiektu, odblasków, kształtu (konkretnie kształtu jego powierzchni) lub przezroczystości. Wszystkie te elementy uzyskuje się nakładając tekstury w sposób płaski, walcowy bądź kulisty.

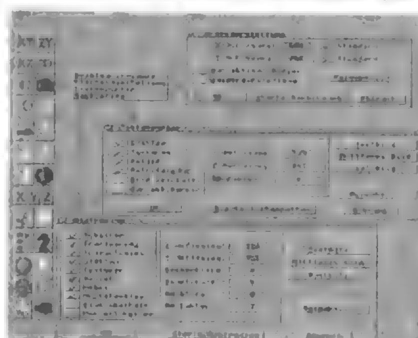
Przygotowane sceny mogą być oświetlane globalnie lub miejscowo ze specjalnie ustawionych źródeł o promieniowaniu rozchodzącym się w sposób kulisty, stożkowy lub walcowy (podaje się wówczas odpowiednie parametry takie jak: kąt rozwarcia stożka światła, barwa światła, jego zasięg itp.). Również obiekty same w sobie mogą być źródłami oświetlenia.

Ustawiając odpowiednie preferencje, można uwidocznić na rzutniach takie elementy jak: stożek widzenia kamery, zasięg i kierunek oświetlenia, aktualne nazwy obiektów, itp. Dzięki temu daje się w łatwy sposób "zapanować" nad najbardziej nawet skomplikowanymi scenami.

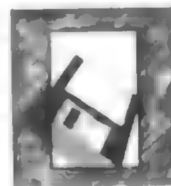
Gdy już wszystkie elementy zostały odpowiednio przygotowane, wówczas można zająć się właściwym tworzeniem rysunków. "Maxon Cinema 4D" oferuje cztery tryby przeglądania scen: "Drahtdarstellung", "Flächendarstellung", "Tiefenpuffer" oraz "Raytracing". W pierwszym przypadku możemy obejrzeć "szkieletowe" przedstawienie naszej sceny. Drugi tryb uwzględnia kolor przedmiotu oraz kierunek i natężenie



Opcje związane z atrybutami.



Różne sposoby generacji scen.



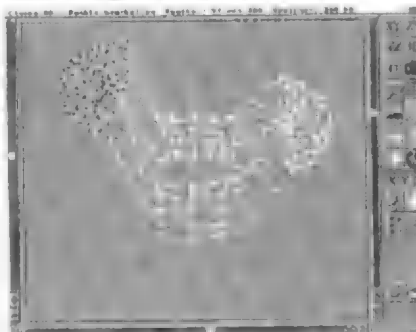


Ekran roboczy programu Maxon Cinema4D.

oświetlenia. W "Tiefenpuffer" pojawiają się dodatkowe możliwości, ponieważ zostają tu uwzględnione atrybuty i tekstury nadane powierzchniom obiektów. Ostatnia opcja pozwala uzyskać najlepszy jakościowo obraz, ale wymaga dużej dozy cierpliwości. Poza pierwszym trybem mamy możliwość korzystania [redacted] wszystkich standardowych rozdzielczości (do 1024*768 punktów), zarówno w HAM-6, HAM-8, jak i w grafice 24-bitowej.

Jedyna, naprawdę poważna wada programu jest związana z nadmiernym uproszczeniem możliwości animacyjnych ograniczających się do sterowania położeniem kamery. Takie potraktowanie sprawy skutecznie utrudnia pracę, a uzyskanie nawet stosunkowo prostych efektów wymaga dodatkowych zabiegów i sztuczek.

Samo tworzenie animacji może się odbywać skokowo, wówczas samodzielnie przesuwamy kamerę przy każdej następnej klatce lub w sposób ciągły. Przy drugim sposobie definiuje się krzywą wzdłuż której przesuwa się kamera (wskazując kolejne, charakterystyczne położenia kamery), liczbę klatek animacji [redacted] sekundę oraz szereg innych, niezbędnych parametrów. Tak utworzone klatki łączy się w jedną całość za pomocą znajdujących się w zestawie programików (np. "AnimBuild"). Szybkość i sposób generacji rysunków oraz animacji odpowiada temu do czego przyzwyczaił nas "Real 1.4". Podobnie jak i w tamtym programie mamy możliwość bezpośredniego "podglądania" tworzonej właśnie kompozycji (program rysuje wówczas linię po linii, [redacted] następnie kolejne klatki animacji). Jest to niezwykle przydatne, przy braku opcji pozwalającej na wygenerowanie tylko zaznaczonego obszaru sceny, ponieważ pozwala na szybką weryfikację ustawionych parametrów (atrybuty i tekstury nadane obiektom, sposób oświetlenia, itp.). Obrazki mogą być tworzone w pamięci komputera lub przy większych rozdzielczościach z 24-bitową paletą kolorów,



Program pozwala na edytowanie obiektów bezpośrednio w oknie Perspective.

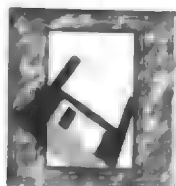
bezpośrednio na dowolny zewnętrzny nośnik pamięci (dysk twardy, dyskietkę).

Podsumowując, "Maxon Cinema 4D" jest udanym programem, w pełni spełniającym postawione przed nim zadania. Za bardzo niską cenę uzyskuje się pełen system pozwalający na zupełnie przyzwoitą pracę na dowolnym modelu Amigi (oczywiście im więcej pamięci tym lepiej). Za największy "plus" programu można uznać przejrzysty i niezwykle szybki edytor, ze szczególnym uwzględnieniem dużej łatwości w dostępie do wszystkich opcji programu. To ostatnie związane jest z przypisaniem osobnych funkcji zarówno lewemu, jak i prawemu przyciskowi myszy. Lewy klawisz odpowiedzialny jest bowiem za bezpośrednie wywołanie danej opcji, natomiast prawy powoduje przejście do specjalnych okienek konfiguracyjnych, w których można ustawić określony sposób pracy danej funkcji (np. przy tworzeniu obiektów będą to: liczby przekrojów poziomych, liczby ścianek, kąty obrotu, itd.). Jeżeli firma "Maxon" postara się o wydanie kolejnej wersji "Cinemy 4D", w której zostaną rozbudowane opcje dotyczące animacji, wówczas program ten stanie się podstawowym narzędziem dla posiadaczy "mniejszych" Amig.

PS. Nie wspominałem o tym wcześniej, ale pewną tradycją firmy Maxon jest to, iż najpierw wydaje ona swoje programy w wersji niemieckojęzycznej, a dopiero po przyjęciu się produktu [redacted] rynku, tworzy inne wersje w zależności od zapotrzebowania. Jeżeli więc ktoś "nie cierpi" języka niemieckiego, będzie musiał uzbroić się w cierpliwość, czekając na wersję angielską programu (oby to tylko nie trwało tak długo, jak w przypadku "Maxon Cada"). ❑



Przykład możliwości programu.



SCALA MM 300

0 10 20 30 40 50 60 70 80 90 100 110 120 130 140 150

Krzysztof [illegible]

Codziennie ponad [illegible] milionów ludzi na całym świecie ogląda prezentacje wykonane tym programem. Stosowany jest w profesjonalnych studiach Betacam jako wyśmienity tani generator napisów. Jest niezbędnym narzędziem w studiach telewizji regionalnych i kablowych.

Pod koniec roku 1993 ukazała się nowa wersja popularnego programu Scala Multimedia z oznaczeniem MM300. Główną przyczyną ukazania się tej wersji był program Media Point, który ze swoimi dodatkami znacznie przewyższał starą wersję Scali MM200.

Co nowego w Scali Multimedia MM300?

AGA

Nareszcie możemy w pełni wykorzystać zalety kości AGA. Program umożliwia tworzenie stron w 256 kolorach w max. rozdzielczości 1472x566 (Super HI-res).

Rozszerzone opcje systemowe.

Można bezpośrednio z menu SYSTEM wywołać takie programy jak Fix Script, Scala Print, Scala AnimLab. Automatycznie pojawiają się w menu programy nowe, umieszczone w katalogu Utilities:

Istnieje możliwość zmiany trybu wyświetlania prezentacji. Poprzez każdorazowe kliknięcie na menu PLAYBACK wybieramy tryb w jakim ma być wyświetlana prezentacja (NTSC, PAL, DOUBLE PAL, MULTISCAN, DOUBLE NTSC):

Zmiana trybu wyświetlania Menu (MENUS):

Możemy wybrać sposób powtarzania wyświetlania skryptu - określoną ilość razy i nieskończenie wiele:

Przy tworzeniu systemów Multimedialnych Interaktywnych możemy wybrać sposób sterowania prezentacją - za pomocą myszy lub joystika.

Łatwość prezentacji statystycznych.

Scala MM300 ma dodane nowe funkcje, które umożliwiają łatwe tworzenie linii, okręgów, kwadratów i prostokątów. Każda figura może być wypełniona kolorem lub być przezroczysta z kolorową obwódką. Funkcja 3D PLUS umożliwia dodanie do prostokątów, kwadratów i linii efektu wypukłości.

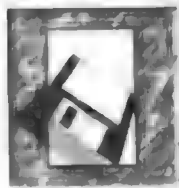
Skalowanie pędzli (brush).

Każdy wgrany brush możemy skalować tak jak w programach DTP. Do każdego brusha możemy dodać kolorową obwódkę (outline), cień (shadow).

Floyd - Steinberg.

Scala w czasie rzeczywistym stosuje na wszystkich brushach technikę ditheringu, która w maksymalnym stopniu odtwarza kolory pędzli wykorzystując paletę podkładu. Np. rysunek będący naszym podkładem jest tylko w 16 kolorach, chcemy użyć pędzla (brusha) w 32 kolorach. Bez ditheringu Floyda - Steinberg brush przyjąłby kolory pod





kładu i wyglądałby naprawdę żałośnie. Metoda Floyda-Steinberga polega na umieszczeniu blisko siebie wielu punktów o różnych kolorach. Patrząc z daleka na taki brush mamy złudzenie, że nie brakuje mu żadnego z kolorów (dzięki małej rozdzielczości naszego wzroku widzimy więcej kolorów niż jest w rzeczywistości):

Optymilizacja palety kolorów.

Stosując funkcję Optimize Palette program likwiduje z palety kolory nie używane na danej stronie graficznej.

Zmiana rozdzielczości podkładu.

Scala MM300 umożliwia zmianę rozdzielczości dowolnego podkładu (background). Nową rozdzielczość wpisujemy bezpośrednio z klawiatury lub wybieramy dowolną już zdefiniowaną.

Opcje dźwiękowe.

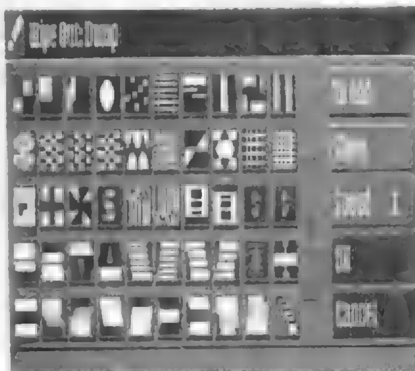
Można zdefiniować oddzielne dźwięki dla klawiszy oznaczonych i klawiszy wyboru. Np. gdy najedziemy w czasie prezentacji interaktywnej na jakieś pole wyboru to zmieni ono swój kolor i wyda określony dźwięk, jeżeli klikniemy na to pole to wyda on inny dźwięk oznaczający wybór.

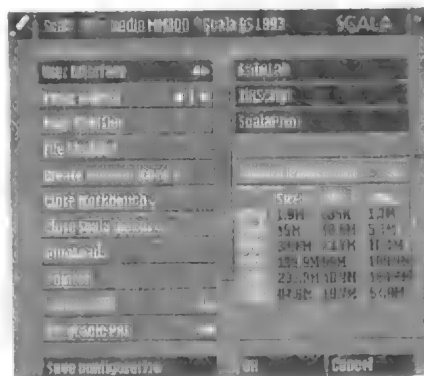
Synchronizacja MIDI.

Scala wykorzystuje absolutnie nowy system synchronizacji dźwięku z wykorzystaniem wejść portu MIDI.

Nowe efekty tekstowe.

Nareszcie Scala MM300 umożliwia stosowanie podkreślenia tekstu w innym kolorze. Np. cały tekst może być napisany kolorem żółtym, a podkreślony czerwoną linią. Dla lepszej czytelności tekst może być umieszczony na





polach z wykorzystaniem rastrowania (background box). Tekst może być przesuwany bezpośrednio na stronie graficznej bez naciskania opcji MOVE. Dodano nowe efekty ukazywania się tekstu i (obsolutna nowość) zanikania.

Optymilizacja programu

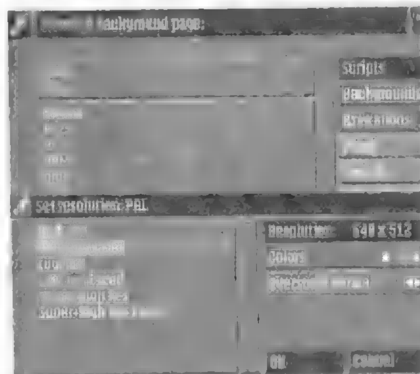
Scala MM300 wykorzystuje do pracy przede wszystkim pamięć FAST. Bez tej pamięci praca programu jest nie możliwa. Wszystkie efekty działają na Amidze z 1 MB Chip Ram i 6 MB Fast Ram. Program w porównaniu z wcześniejszymi wersjami działa znacznie szybciej. Długość wgrywanych efektów dźwiękowych, jak i samej muzyki jest tylko ograniczona wielkością twardego dysku.

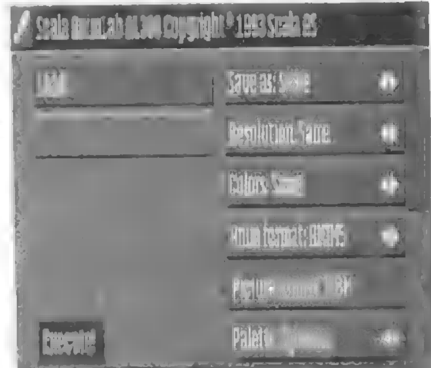
Antialiasing

Nowa Scala dla super wygładzenia schodkowatych krawędzi stosuje 4 poziomowy antialiasing.

AnimLab AL300

Nowa wersja programu AnimLab umożliwia konwersję animacji na dowolną rozdzielczość. Jest dodana optymilizacja kolorów z użyciem ditheringu Floyda-Steinberga. Program wy-





korzystuje nowe formaty animacji m.in Anim-7.

Narzędzia

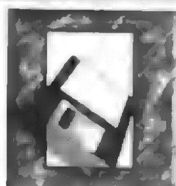
Za pomocą dołączonych programów możemy wydrukować dowolny skrypt w postaci tekstowej lub graficznej. Program wykorzystuje wszystkie systemowe drivery drukarek, PostScript i kości AGA. Przy każdym nagrywaniu skryptu na dysk Scala MM300 pyta się, czy nagrać skrypt wraz z wykorzystywanymi podkładami, fontami. Jest to bardzo pomocna funkcja przy przenoszeniu skryptu na inny twardy dysk.

EXes

Do Scali MM300 dołączono programy, które obsługują nowe urządzenia zewnętrzne:
- SuperGen
- G-Lock
- G2 VideoCenter
- Impact Vision 24
- AVideo
- Sunrize Studio 16
- NewTek Video Toaster

Dodano nowe wersje MIDI 2.0 i CDTV 2.0 oraz drivery do obsługi sprzętu: Sony, Philips, Pioneer Laserdisk, Canon Stillvideo, Link, DVE-10, Panasonic AG5700, AG7330, AG7350, Sony U-Matic, Mitsubishi E82.

Najnowsza wersja Scali jest naprawdę w pełni profesjonalnym produktem o dużych możliwościach. Nareszcie można wczytywać spakowane popularnym Power Packerem rysunki, teksty i moduły muzyczne. Po dołączeniu odpowiednich driverów można wczytywać bezpośrednio pliki w formacie BMP, PCX, TIFF, GIF, FLI, FLC. W jednym z najbliższych numerów Amigowca zajmę się Scalą od strony nieco bardziej użytkowej. ❑



IMAGINE v2.9 i co dalej?

Marek Sing

Po długim oczekiwaniu dotarła do mnie nowa wersja Imagine. Niestety, nie jest to wersja 3.0, tak dawno zapowiadana przez firmę Impulse. Imagine 2.9 w stosunku do wersji 2.0 charakteryzuje się niewielkimi, wręcz kosmetycznymi zmianami. Oczekiwany skok jakościowy nie nastąpił.

Program otrzymujemy na dwóch dyskietkach. Na jednym dysku jest wersja na komputer z kartą turbo i koprocesorem, na drugim wersja na każdą Amigę. Programy na dysku są zarchiwizowane i trzeba je rozpakować na dysk twardy lub na sformatowany w FFS dysk!? Plik główny ma wielkość ponad 850KB i posiadacze systemu 1.3 bez twardego dysku mogą zapomnieć o nowym Imagine. Także praca z jednym lub nawet dwoma megabajtami pamięci może okazać się niemożliwa. Po uruchomieniu programu pojawia się tytułowa plansza z napisami Imagine v.2.9 Under Construction. Ten napis na dole wyjaśnia wszystko. Program jaki otrzymujemy nie jest gotowym produktem komercyjnym przeznaczonym do sprzedaży, a tylko wersją przeznaczoną do testowania (chociaż normalnie sprzedawaną). Podejrzewam, że Impulse zdecydował się na takie rozwiązanie, aby nie stracić swoich dotychczasowych wielbicieli, zniecierpliwionych oczekiwaniem na nową wersję i nowe możliwości Imagine, na rzecz Reala 2.0 lub Lightwave'a.

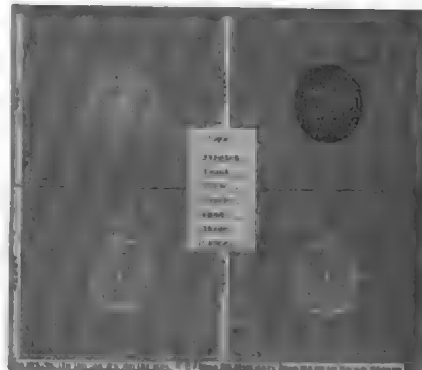
Co nowego w wersji 2.9?

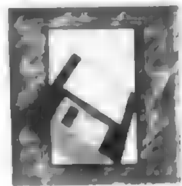
Wygląd ekranów poszczególnych edytorów nie uległ zmianie. Pojawiły się tylko nowe opcje i jeden nowy edytor (Spline Editor). Posiadacze Amigi 1200 i 4000 mogą skorzystać z nowych trybów renderowania dostosowanych do kości AGA. Niestety, w nowych trybach nie można wygenerować projektu w postaci animacji w jednym pliku, trzeba generować pojedyncze klatki i łączyć np. przy pomocy Scali.

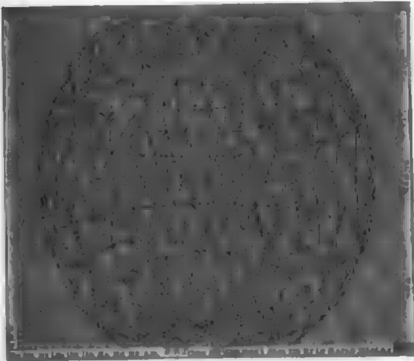
Najwięcej zmian dokonano w oknie perspektywy. Jest ono teraz wygodniejsze w obsłudze i bardziej przejrzyste. Dodano trzy przyciski do obracania, powiększania i zmniejszania obiektu, a także możliwość ustawiania perspektywy oglądanego obiektu. Przy pomocy myszki możemy obracać obiekt i manipulować nim w czasie rzeczywistym. Możemy skorzystać z funkcji Reset, która powoduje powrót do ustawionych standardowo parametrów widoku. Pojawił się nowy tryb oglądania w oknie perspektywy, zwany NewMode. Kiedy uruchomi się nowy tryb w oknie podglądu zostaną pokazane wszystkie osie, punkty, itp., czyli wszystko, co znajduje się w innych oknach. Jest to bardzo pomocna funkcja przy pracy w Stage Editorze, ponieważ można ustawiać obiekty w oknie Camera View (przesuwa się obiekty tak, jakby patrzyło się przez kamerę). W Detail Editor dodano kilka nowych funkcji do obróbki i deformacji obiektów. Możemy tu znaleźć takie funkcje jak rozciąganie obiektu, skręcanie, zwężanie obiektu ku końcowi, ugniatanie, zginanie i wycinanie. Dodano także możliwość wczytywania i zapamiętywania obiektów w standardzie DXF, popularnym na peceta. Poprawiono pracę przy budowaniu i przerabianiu obiektów. Quick Edges pozwala na obejrzenie w oknie perspektywy tylko zaznaczonych brzegów. Na uwagę zasługuje opcja Latticizie, która pozwala budować obiekty z samych szkieletów. Bardzo przydatna rzecz przy tworzeniu animacji z zastosowaniem przekroju obiektu.

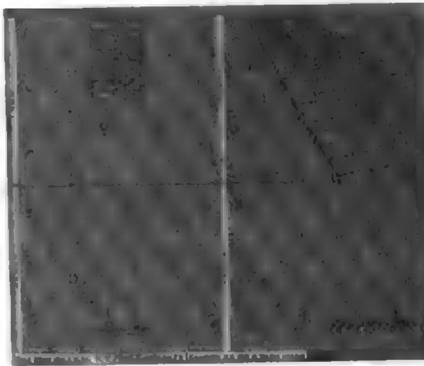
Jedną z nowych rzeczy zastosowanych w Stage Editorze są linie pokazujące pole widoku jaki obejmuje kamera lub lampa kierunkowa. Nareszcie kończą się problemy z ustawianiem i obracaniem kamery. Zastosowanie opcji NewMode pozwala, w oknie widoku









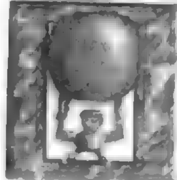




kamery, przy użyciu myszki przesuwać, obracać i skalować obiekty [redacted] czasie rzeczywistym .

Spline Editor ładuje fonty w formacie PostScript Type 1 i automatycznie tworzy obiekty trójwymiarowe, pościankowane z wytłoczeniem i drugą stroną. Niestety nie przyjmuje on Compugraphic Fonts, które stały się standardowyn formatem Amigi. Imagine 2.9 oferuje około 100 nowych tekstur do wykorzystania, ale nie przyjmuje starych tekstur np. Essence. W ten sposób starsze obiekty wykonane na Imagine 1.1 lub 2.0 i obłożone teksturami [redacted] są kompatybilne z Imagine 2.9. Z nowych tekstur najładniejsza jest Lens Flare, która pozwala na wykonanie efektu np. wybuchu gwiazdy. Udało mi się znaleźć tylko jeden nowy efekt, [redacted] pomocą którego możemy symulować wiatr, grawitację, śnieg, deszcz (trochę mało jak na nową wersję). Nowy Imagine nie przyjmuje także atrybutów wykonanych na starszych wersjach. Tak samo programiści nic nie zrobili [redacted] temacie anty-aliasingu (czyli wygładzania obiektów) i daleko mu jeszcze do jakości Reala 2.0.

Podsumowując, Imagine v.2.9 nie jest tym na co wszyscy czekają. Niewielkie zmiany w stosunku do Imagine 2.0 nie wnoszą do programu nic nowego. Pożyjemy zobaczymy. Może firma Impulse obudzi się z letargu i pokaże program na miarę swoich możliwości. ❑



BRILLIANCE

Profesjonalny program graficzny cz.4

[illegible]

Powoli dobiega końca opis podstawowych możliwości Brilliance'a. Nie zdążyliśmy się jeszcze w pełni do niego przyzwyczaić, a już pojawiły się sygnały o nowej wersji tego programu, w której m.in. mają zostać dodatkowo rozbudowane opcje dotyczące animacji. Pozostańmy jednak przy omawianej wersji 1.0. Przewodnim tematem naszego ostatniego spotkania będzie zasygnalizowane już wcześniej menu Tweening.

Pod nazwą tą kryje się wspaniałe narzędzie pozwalające na tworzenie dowolnych rysunków i animacji z pełnym zachowaniem specjalnie ustawianej perspektywy. Można powiedzieć, że **"Tweening menu"** spełnia dwa podstawowe zadania. Pierwsze z nich związane jest z przekształcaniem wyciętych brushy, poprzez zmianę ich położenia lub obrót względem osi X,Y,Z. Pozwala to na łatwe uzyskanie wrażenia trójwymiarowości tworzonej grafiki. Tak przygotowane brushe można wykorzystać jako fragment rysunku lub animacji. Program pozwala na wypełnianie całego ekranu wyciętym "pędzlem", w myśl ustawionych parametrów perspektywy. Druga rola **"Tweening menu"** jest związana z ułatwianiem procesu przygotowywania animacji, pozwalającym na jej tworzenie poprzez podanie jedynie charakterystycznych położeń animowanego elementu, podczas, gdy wszystkie etapy pośrednie zostają automatycznie wypełniane przez komputer.

Przejdźmy teraz do wspomnianego menu. Znajduje się ono w dolnym rzędzie przycisków panelu głównego jako trzecia ikonka od lewej strony. Po jej wskazaniu pojawi się rozbudowane okno, którego funkcje są ściśle powiązane z innymi elementami programu. Przedstawienie **"Tweening menu"** rozpocznę od lewej strony. I tak:

- ***"Copy start to end"*** - powoduje skopiowanie parametrów ustawienia brusha z położenia początkowego do położenia końcowego animacji lub wypełniania ekranu,
- ***"Copy end to start"*** - analogicznie do poprzedniego, przy czym skopiowane zostaje ustawienie końcowe w miejsce ustawienia początkowego (pozwala to na łatwe tworzenie animacji poprzez "sklejanie" ze sobą kolejnych jej fragmentów),
- ***"Swap start with end"*** - zamienia między sobą ustawienia "Start" i "End",
- ***"Get last brush pos."*** - ustawieniu "Start" lub "End" zostaną przypisane parametry odpowiadające położeniu na ekranie ostatnio naniesionego brusha,
- ***"Reset values"*** - przywraca parametry "zerowe" dla odpowiedniego ustawienia.

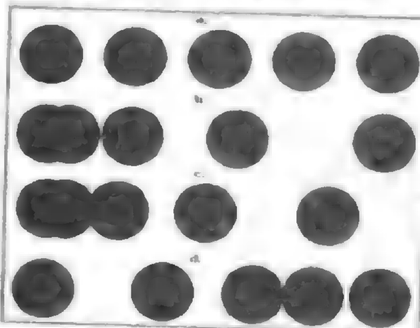
Pewnego uściślenia wymagają dwa użyte powyżej sformułowania: ustawienie i parametr. Pod pierwszym kryje się początkowe ("Start") lub końcowe ("End") położenie brusha powiązane z tworzeniem animacji, bądź wypełnianiem ekranu. Natomiast parametrem będą odpowiadające określonym położeniom wartości liczbowe przypisane pozycji i obrotowi brusha.

Informacje o aktualnych parametrach znajdują się tuż obok wspomnianych wcześniej opcji, są to odpowiednio:

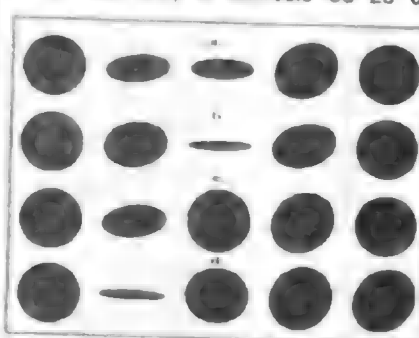
- ***"Position X,Y,Z"*** - współrzędne położenia brusha,
- ***"Rotation X,Y,Z"*** - parametry obrotu,

Znajdują się tu również:

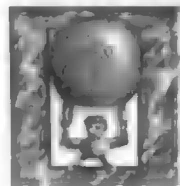
- ***"Opacity"*** - w okienku tym podaje się liczbową wartość (od 0 do 100) określającą stopień przezroczystości rysowanych elementów (podanie różnych wartości dla opcji "Start" i "End" spowoduje stopniowaną zmianę prześwitywania),
- ***"Start"*** - wyświetla aktualne parametry ustawienia początkowego,
- ***"End"*** - analogicznie jak poprzednio, przy czym zostają podane parametry ustawienia końcowego,
- ***"Adjust"*** - dodatkowe menu pozwalające na swobodną zmianę szeregu parametrów bezpośrednio na ekranie za pomocą myszy. Można w nim dowolnie zmieniać położenie obiektu, zarówno w płaszczyźnie rysunku (czyli wzdłuż osi X,Y), jak i w płaszczyźnie prostopadłej do ekranu (oś Z). To ostatnie powoduje zmniejszanie lub powiększanie brusha, odpowiadające jego przybliżaniu i oddalaniu się od obserwatora zgodnie z zasadami perspektywy. Mamy możliwość dowolnego obracania elementu dookoła osi X,Y,Z. Niezwykle przydatnymi opcjami są "Lens" i "Orig.". Pierwsza z nich pozwala na zmianę ogniskowej (podobnie jak w obiektywach) w zakresie od 25 do



Rysunek 1.



Rysunek 2.



2000. Natomiast druga wyznacza kierunek "zbiegania się" perspektywy na naszym rysunku.

Następnym oknem jest **"Brush relative"**, [redacted] w nim:

- ***"Position relative"*** - powodujący włączenie względnego układu współrzędnych, związanego z początkowym położeniem brusha. W układzie takim współrzędne są mierzone od parametrów ustawienia początkowego brusha, [redacted] nie od środka perspektywy.

- ***"Rotate relative"*** - działa podobnie jak poprzednia opcja, przy czym dotyczy obrotów brusha.

Pod **"Brush relative"** znajdują: ***"Go back", "Cyclic"*** oraz ***"Aspect"***. W połączeniu z innymi opcjami **"Tweening menu"** (np. ***"Fill", "Trail length"***) pozwalają one uzyskać różnego typu sposoby wypełniania ekranu. Przykłady zastosowania tej opcji można znaleźć na rysunkach 1 i 2. Odpowiednio będą to:

-dla rys. 1 Pos. [redacted] settings-X:

-a- ustawienie standardowe,

-b- "Go back" + "Trails" + "Start ease rate=99" + "Move anim forward",

-c- "Cyclic" + "Trails" + "Start ease rate=99" + "Move anim forward",

-d- "Aspect" + "Trails" + "Start [redacted] rate=-20" + "XOver=50%" + "Move anim forward",

-dla rys. 2: Rot. ease settings-X:

-a- ustawienie standardowe,

-b- "Go back" + "Trails" + "Start ease rate=60" + "Move anim forward",

-c- "Aspect" + "Trails" + "Start [redacted] rate=99" + "XOver=50%" + "Move anim forward",

-d- "Aspect" + "Trails" + "Start [redacted] rate=-99" + "XOver=90%" + "Move anim forward".

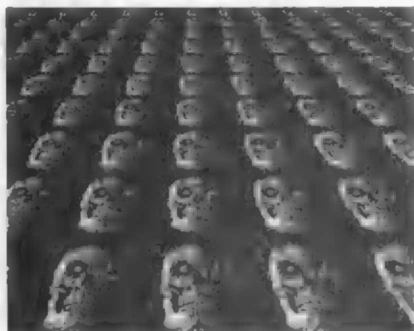
Dalej znalazły się:

- ***"Fill [redacted] with brush"*** - wypełniający cały ekran wyciętym brushem z zachowaniem odpowiednich parametrów perspektywy,

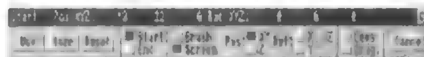
- ***"Create [redacted] brush trail"*** - rysuje kolejne położenia "pędzla", zawarte pomiędzy ustawieniem początkowym [redacted] końcowym.

Przyciski:

- ***"Move anim backwards"*** - w przypadku tworzenia animacji kolejne położenia brusha będą rysowane od końca,



Przykład zastosowania perspektywy.



Tweening menu-Adjust.

- ***"Pause animation"*** - wszystkie położenia znajdą się na jednym rysunku,

- ***"Move anim forward"*** - podobnie jak "Backwards", przy czym kolejne położenia będą rysowane od początku.

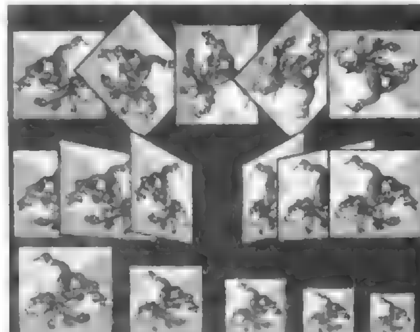
Prawą część **"Tweening menu"** zajmują informacje dotyczące: liczby kroków, w jakich ma nastąpić przejście pomiędzy początkowym, a końcowym położeniem brusha ("Number of steps"), początku ("Start draw step number") i końca tego przejścia ("End draw step number") oraz liczby powtórzeń kolejnych elementów ("Trail length").

Pozostały jeszcze do omówienia dwie opcje:

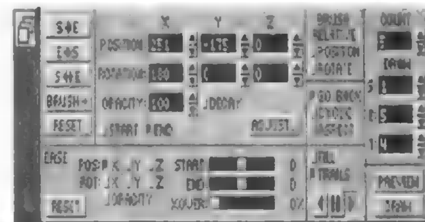
- ***"Preview"*** - odtwarza utworzone przejście w uproszczeniu, zastępując brusha ramką odpowiedniej wielkości. Ramka ta odzwierciedla zarówno ruch postępowy jak i obrotowy brusha.

- ***"Draw"*** - uruchamia proces wypełniania ekranu.

Jak stworzyć animację za pomocą **"Tweening menu"**? W pierwszej kolejności musimy w **"Animation menu"** zadeklarować interesującą nas liczbę klatek animacji. Następnie wycinamy fragment obrazu, który ma się przemieszczać po ekranie. Przechodzimy do **"Tweening menu"** i za pomocą parametrów ***"Position X,Y,Z"*** oraz ***"Rotation X,Y,Z"*** (lub bezpośrednio na przygotowywanym rysunku za pomocą opcji ***"Adjust"***) określamy położenie początkowe (**"Start"**) i końcowe (***"End"***) brusha. Jeżeli wybierzemy teraz opcję ***"Trails"***, a następnie ***"Draw"***, wówczas program samoczynnie utworzy następujące po sobie przemieszczenia brusha (zgodnie z zadeklarowaną ilością kroków - ***"Steps"***), umieszczając je w kolejnych klatkach animacji. Bardziej złożone animacje tworzy się poprzez "sklejanie" ze sobą szeregu jej fragmentów. Wykorzystuje się do tego celu opcję ***"Copy end to start"*** pozwalającą na zmianę kierunku ruchu wyciętego "pędzla", poprzez przypisanie



Tak mogą wyglądać kolejne klatki animacji.



Tweening menu.

końcowych parametrów poprzedniego przemieszczenia w miejsce nowego ***"Startu"***. Nadając następnie nowe parametry opcji ***"End"*** uzyskujemy kolejną "ścieżkę", po której będzie się poruszał brush. W połączeniu z możliwościami omówionymi w poprzednich odcinkach opisu, pozwala to [redacted] praktycznie dowolną manipulację obrazem.

Na zakończenie opisu "Brilliance'a" chciałbym zwrócić uwagę Czytelników na kilka istotnych rzeczy związanych z pracą [redacted] tym programem (i nie tylko):

- "Brilliance" otwiera własny, specyficzny ekran, [redacted] może stać się przyczyną pewnych nieprawidłowości w jego działaniu zależnie od skonfigurowania systemu. Można tego uniknąć, uruchamiając przed głównym programem znajdujące się w katalogu "C" programiki: "SetPatchMrgCop" oraz "SetPatch",

- chcąc uniknąć kłopotów z brakiem pamięci należy używać tylko takiej ilości kolorów, która jest niezbędna dla uzyskania porządanego efektu (szczególnie w przypadku animacji komputerowej),

- można dodatkowo zaoszczędzić RAM "wyłączając" system oraz zmniejszając lub całkowicie usuwając bufor przeznaczony dla opcji "Undo" (preferencje programu).

Podsumowując nasze spotkania z "Brilliancem" można stwierdzić, iż jego pojawienie się zmieniło filozofię i jakość obsługi programów graficznych. Jeżeli zapowiadany od dawna "Deluxe Paint 5.0" nie będzie zdecydowanie szybszy od swoich poprzedników, wówczas "Brilliance" ma szansę stać się niepodzielnym "królem barwy i ruchu" Amigi, ustępując jedynie profesjonalnemu oprogramowaniu kart graficznych.

P.S. Starając się przedstawiać poszczególne możliwości programu [redacted] pewnych blokach tematycznych (rysowanie, obróbka brushy, animacja) umknęła mojej uwadze jedna z opcji głównego panelu, mianowicie "Fast feedback". Pozwala ona na przyspieszenie pracy [redacted] przypadku rysowania figur niestandardowym pędzlami. Kształt, którym będzie kreślona linia, pojawia się wówczas tylko na jej początku i końcu, podczas, gdy cała reszta rysowana jest normalną, cienką linią. I to już naprawdę wszystko. ❑



KURS ASEMBLERA cz.7

Adam Gregorowicz

Ostatnio poznaliście instrukcje arytmetyczne mikroprocesora Motorola 68000, dzisiaj - zgodnie z zapowiedzią - rozpoczniemy pisanie programu prostego kalkulatora. Jest to dosyć trudne zadanie, ale przekonacie się, że jesteście w stanie mu sprostać.

Zanim jednak zajmiemy się programem kalkulatora winien Wam jestem małe wyjaśnienie. Otóż już dwukrotnie w listingach pojawiła się nie wyjaśniona wcześniej instrukcja LEA. LEA jest na szczęście bardzo prostą do zrozumienia instrukcją i służy tylko i wyłącznie do wpisywania wartości (adresu) do rejestrów adresowych.

LEA - Load effective address. (Wpisanie adresu efektywnego.)

Zapis: LEA źródło,An

Instrukcja LEA wpisuje adres do określonego rejestru adresowego. Zmieniane są zawsze wszystkie 32 bity rejestru adresowego. Z reguły instrukcję LEA można zastąpić instrukcją MOVEA (np. instrukcja LEA GraphBase,A0 wykonuje dokładnie to samo, co MOVEA.L #GraphBase,A0). Jednak rozkaz LEA umożliwia stosowanie trybów adresowania niedostępnych dla MOVEA (np. tryb adresowania z przesunięciem względem licznika programu - LEA adres(PC),An).

Stosując rozkaz LEA można również dodać stałą do rejestru adresowego bez zmieniania znaczników (można go więc stosować zamiennie z instrukcją ADDA).

Znaczniki:

X - Nie zmieniany.
N - Nie zmieniany.
Z - Nie zmieniany.
V - Nie zmieniany.
C - Nie zmieniany.

Dla rozkazu LEA dozwolone są następujące tryby adresowania:

LEA (An),Am
LEA p(An),Am
LEA p(An,Dm.W),Ar
LEA p(An,Dm.L),Ar
LEA adr.W,An
LEA adr.L,An
LEA p(PC),An
LEA p(PC,Rn.W),Am
LEA p(PC,Rn.L),Am

Przykład: LEA $70000,A0 powoduje wpisanie do rejestru adresowego A0 liczby $70000. Proszę zwrócić uwagę, że do rejestru A0 jest wpisywana LICZBA $70000. Rozkaz MOVEA.L $70000,A0 spowodowałby wpisanie do A0 ZAWARTOŚCI komórki pamięci o adresie $70000. Dopiero rozkaz MOVEA.L #$70000,A0 wykonałby tę samą operację co LEA $70000,A0.

Przykład: LEA 20(A1),A2 powoduje wpisanie do rejestru A2 zawartości rejstru A1 powiększonego o 20.

Przykład: LEA 100(A0,D0.L),A1 powoduje wpisanie do rejestru adresowego A1 zawartości rejestru A0 powiększonej o zawartość rejestru danych D0 i liczbę 100.

Mam nadzieję, że teraz już każdy będzie wiedział co robi np. instrukcja LEA GraphicsName(pc),a1. Skoro tak, to możemy przejść do pisania programu kalkulatora.

Jak każdy dobry program powinien on umożliwiać obsługę za pomocą myszki (jest to w końcu domeną Amigi), więc nie obejdzie się bez prześledzenia sposobów korzystania z tak zwanych gadżetów. Na początek przedstawię strukturę Gadget definiującą pojedynczego gadżeta.

Opis struktury Gadget:

+00 długie słowogg_NextGadgetadres definicji następnego gadżetu lub 0 gdy nie ma więcej gadżetów
+04 słowo gg_LeftEdge pozycja lewej krawędzi pola gadżetu względem lewej krawędzi okna
+06 słowo gg_TopEdge pozycja górnej krawędzi pola gadżotu względem górnej krawędzi okna
+08 słowo gg_Width szerokość pola gadżetu
+10 słowo gg_Height wysokość pola gadżetu
+12 słowo gg_Flags określa rodzaj zmiany wyglądu gadżetu po jego naciśnięciu
+14 słowo gg_Activation odpowiada za zachowanie gadżetu po jego zwolnieniu
+16 słowo gg_GadgetTypetyp gadżetu
+18 długie słowogg_GadgetRenderadres struktury Border lub struktury Image (przy gadżecie "obrazkowym")
+22 długie słowogg_SelectRenderzero lub adres stuktury Image przy gadżecie "obrazkowym"
+26 długie słowo gg_GadgetTextadres definicji tekstu połączonego z gadżetem
+30 długie słowo gg_MutualExclude używane przy gadżetach współzależnych
+34 długie słowo gg_SpecialInfoinformacje specjalne
+36 słowo gg_GadgetID słowo do wykorzystania przez programistę
+40 długie słowo gg_UserDatadługie słowo do wykorzystania przez programistę

Pole gg_Flags odpowiedzialne jest za wygląd gadżetu po jego zaktywizowaniu. Najczęściej używane są gadżety, które po wybraniu zmieniają kolor. Uzyskuje się je wpisując w pole gg_Flags wartość GADGHCOMP (odpowiada jej liczba 0).

W polu gg_Activation określa się sposób zachowania się gadżetu po naciśnięciu. Jeśli wybierzemy GADGIMMEDIATE

(wartość 2) i RELVERIFY (wartość 1), to po wciśnięciu przycisku myszy nad gadżetem zmieni on kolor (oczywiście jeśli ustawimy GADGHCOMP w gg_Flags), a po puszczeniu klawisza myszki powróci do normalnego stanu. Jeśli natomiast do tego ustawienia dodamy jeszcze TOGGLESELECT (wartość 256), to po wciśnięciu gadżet zachowa się tak samo (zmieni kolor), lecz po zwolnieniu przycisku myszy pozostanie w tym stanie aż do następnego wybrania. W naszym kalkulatorze użyjemy ustawienia GADGIMMEDIATE+RELVERIFY.

Natomiast w polu gg_GadgetType ustawia się rodzaj gadżetu. Nas będzie interesował najprostszy ich typ - czyli gadżety reagujące na wciśnięcie. Wybiera się je wpisując w pole gg_GadgetType wartość BOOLGADGET.

Ważnym elementem większości gadżetów jest ramka (na ogół prostokątna) wskazująca aktywne pole gadżetu. Do jej zdefiniowania używane jest pole gg_GadgetRender, w które wpisuje się adres struktury Border opisującej ową ramkę.

Struktura Border ma następującą postać:

+00 słowo bd_LeftEdge współrzędna lewej krawędzi ramki względem lewej krawędzi gadżetu
+02 słowo bd_TopEdge współrzędna górnej krawędzi ramki względem lewej krawędzi gadzetu
+04 bajt bd_FrontPen kolor linii
+05 bajt bd_BackPen kolor tła (niezbyt użyteczny)
+06 bajt bd_DrawMode tryb rysowania
+07 bajt bd_Count ilość punktów w definicji linii ramki
+08 długie słowo bd_XY adres tabeli zawierającej współrzędne punktów
+12 długie słowo bd_NextBorder adres następnej struktury Border (lub 0, gdy nie chcemy już więcej ramek)

Bliższego omówienia wymaga pole bd_XY i ściśle z nim związana tablica współrzędnych punktów definiujących linie ramki. Tablica współrzędnych (której adres należy wpisać w pole bd_XY) składa się z podanych w rozmiarze słowa współrzędnych (kolejno X i Y) punktów, do których rysowana jest linia. Linia jest zawsze rysowana od ostatnio wyznaczonego punktu do następnego z zaznaczeniem, że pierwszy punkt w tabeli jest początkiem pierwszej lini, a nie jej końcem (gdyż wówczas linia byłaby rysowana nie wiadomo odkąd). W tabeli powinno znajdować się tyle par liczb (współrzędnych X i Y), ile podano w polu bd_Count struktury Border.

Zauważmy, że wszystkie linie zdefiniowane w jednej strukturze Border są rysowane tym samym kolorem (określonym w polu bd_FrontPen) i jeśli chcemy część linii narysować jednym kolorem, a część innym (aby np. uzyskać trójwymiarowy wygląd gadżetów), to trzeba zdefiniować kolejną strukturę Border i połączyć ją z pierwszą wpisując jej adres w polu bd_NextBorder pierwszej struktury.

Aby na przykład otrzymać pseudo trójwymiarowy wygląd przycisków w naszym kalkulatorze należy zastąpić fragment listingu kalkulatora od linii z etykietą "Ramka1" do etykiety "Tekst0" (lini z tą etykietą nie należy już zmieniać) następującą sekwencją:

```
Ramka1:  dc.w  0,0
    dc.b 2          ; drugi kolor z palety workbencha (w systemie 2.0 biały)
    dc.b0
    dc.b0
    dc.b3
    dc.l LineTab1       ; linie rysowane na biało
    dc.l Ramka1B
Ramka1B: dc.w  0,0
    dc.b 1          ; pierwszy kolor z palety workbencha (w systemie 2.0 czarny)
    dc.b0
    dc.b0
    dc.b3
    dc.l LineTab1B      ; linie rysowane na czarno
    dc.l 0
LineTab1: dc.w  24,0
    dc.w  0,0
    dc.w  0,12
LineTab1B: dc.w  0,12
    dc.w  24,12
    dc.w  24,0
```

Prawie każdy gadżet ma wypisany jakiś tekst, do jego zdefiniowania służy pole gg_GadgetText. W polu tym należy wpisać adres struktury IntuiText, która ma następującą postać:

Struktura IntuiText:

+00 bajt it_FrontPen kolor tekstu
+01 bajt it_BackPen kolor tła
+02 bajt it_DrawModetryb rysowania
+03 bajt it_KludgeFill00
+04 słowo it_LeftEdge współrzędna x
+06 słowo it_TopEdge współrzędna y
+10 długie słowo it_ITextFontadres definicji czcionki (jeśli 0, to używany jest font systemowy)
+14 długie słowo it_IText adres właściwego tekstu (tekst ten musi być zakończony bajtem o wartości 0)
+18 długie słowo it_NextText adres nestepnej struktury IntuiText (lub 0)

Bliższego przedstawienia wymaga pole it_DrawMode, w którym należy określić sposób drukowania tekstów. Jeśli wpiszemy tam RP_JAM1 (=0), to rysowane będą jedynie punkty w kolorze rysującym, pozostałe punkty tła pozostaną nienaruszone. Gdy użyjemy RP_JAM2 (=1), rysowane będą zarówno punkty w kolorze rysującym jak i w kolorze tła. W przypadku RP_COMPLEMENT (=2) punkty narysowane na tle pojawiają się normalnie, natomiast nadrukowywanie punktów na już istniejące powoduje ich skasowanie (np. wydrukowanie litery A na istniejącej już takiej samej literze, powoduje jej zmazanie). Przy wybraniu RP_INVERSEVID (=4) wypisywany tekst jest rysowany w kolorze tła, a tło tego tekstu w kolorze ustalonym dla tekstu.

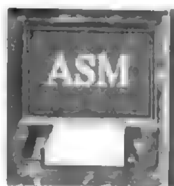
Dla nas najbardziej użytecznym trybem rysowania będzie tryb RP_JAM2.

W listingu na końcu tego artykułu definicje gadżetów rozpoczynają się od etykiety "Klawisz0GAD". Jest tam zdefiniowanych dziesięć gadżetów (dla przycisków kalkulatora od "0" do "9"). Można zauważyć, że w polach gg_GadgetID wpisane są liczby od 0 do 9 (dla każdego gadżetu inna liczba). Do czego one służą? Aby to wyjaśnić muszę opisać co się dzieje jeśli użytkownik wciśnie jakiś gadżet.

System operacyjny Amigi zawiera w sobie coś takiego, co nazywa się IDCMP - Intuition's Direct Communications Messages Ports (w wolnym tłumaczeniu oznacza to ukierunkowane intuicyjnie porty do przesyłania komunikatów). Otóż to IDCMP przez cały czas czeka na jakiekolwiek zdarzenie (może to być włożenie dyskietki do stacji dysków, wciśnięcie jakiegoś gadżetu, przycisku myszy itp.). Po jego wykryciu automatycznie tworzony jest komunikat o tym wydarzeniu. Komunikat ten jest przydzielany do wszystkich portów, które są nim "zainteresowane".

Tak więc cała sztuka polega na otworzeniu własnego portu IDCMP, "zainteresowanie" go zdarzeniem polegającym na wciśnięciu gadżetu i czekanie na pojawienie się komunikatów w tym porcie.

W naszym przypadku otwarcie portu IDCMP jest bardzo proste, gdyż przy otwarciu każdego okienka taki port jest tworzony automatycznie (jego adres można znaleźć w polu



wd_UserPort struktury Window naszego okienka) i dołączany do otwieranego okna. Aby port ten otrzymywał komunikaty dotyczące wciskanych gadżetów należy przed otwarciem okna w pole IDCMPFlags struktury NewWindow wpisać GADGETDOWN (my jeszcze ustawimy znacznik CLOSEWINDOW, aby móc reagować na kliknięcie w gadżet zamykania okienka).

Teraz już możemy czekać na pojawienie się w naszym porcie stosownych komunikatów. Do tego celu użyjemy funkcji WaitPort z biblioteki exec, która wymaga w rejestrze A0 adresu portu IDCMP:

```
_LVOWaitPort equ -384
wd_UserPort equ 86

    move.l  WindowBase(pc),a0   ; adres struktury Window
    move.l  wd_UserPort(a0),a0  ; adres struktury UserPort
                                ; dołączonej do naszego okna
    move.l  4.W,a6              ; baza biblioteki exec
    jsr  _LVOWaitPort(a6)       ; czekanie na sygnał od IDCMP
```

Gdy już wiemy, że jest "do nas" komunikat możemy go odebrać funkcją GetMsg również z biblioteki exec, która tak samo jak WaitPort wymaga w A0 adresu portu IDCMP:

```
_LVOGetMsg equ -372
wd_UserPort equ 86

    move.l  WindowBase(pc),a0
    move.l  wd_UserPort(a0),a0
    move.l  4.W,a6
    jsr  _LVOGetMsg(a6)         ; pobranie komunikatu od IDCMP
```

Jeśli komunikat był poprawny, to w rejestrze D0 otrzymamy adres struktury IntuiMessage zawierającej odebrany komunikat, jeśli coś się nie udało - w D0 będzie zero.

Gdy już mamy odebrany komunikat, należy jak najszybciej odczytać interesujące nas informacje ze struktury IntuiMessage:

```
im_Class  equ 20
im_IAddress equ 28

    move.l  d0,a1               ; wpisanie adresu IntuiMessage do A1
    move.l  im_Class(a1),d2     ; pobranie typu komunikatu
    move.l  im_IAddress(a1),a2  ; pobranie adresu gadżetu,
                                ; który wysłał komunikat
```

Następnie trzeba poinformować system IDCMP, że już odebraliśmy komunikat i w ten sposób pozwolić IDCMP na zajmowanie się obsługą innych komunikatów. Służy do tego funkcja ReplyMsg z biblioteki exec. Potrzebuje ona w rejestrze A1 adresu ostatnio odebranego IntuiMessage:

```
_LVOReplyMsg equ -378

                                ; w a1 jest już adres IntuiMessage
    move.l  4.W,a6              ; baza biblioteki exec
    jsr  _LVOReplyMsg(a6)       ; potwierdzenie odebrania
```

Teraz możemy się zająć interpretowaniem informacji pobranych z odebranego komunikatu. W rejestrze D2 mamy coś, co odczytaliśmy z pola im_Class struktury IntuiMessage, w A2 natomiast zawartość im_IAddress tej struktury. Pole Class to ogólny rodzaj komunikatu (przykładowo - GADGETDOWN informuje, że w aktywnym okienku został wciśnięty jakiś gadżet, lecz nie precyzuje dokładnie który). W polu Class pojawiają się te same wartości, które podaliśmy w polu IDCMPFlags przy definicji okienka. Tak więc gdybyśmy odczytali wartość CLOSEWINDOW znaczyłoby to, że użytkownik wcisnął gadżet zamykania okna. Gdyby zaś pojawiła się wartość GADGETDOWN wiedzielibyśmy, że użytkownik wcisnął jeden z gadżetów połączonych z naszym okienkiem. W tym przypadku musimy skorzystać z informacji z pola im_IAddress. Jest to adres gadżetu, który był źródłem komunikatu (został wciśnięty). Odczytując pole gg_GadgetID tego gadżetu możemy się zorientować, który to gadżet. I dlatego właśnie przy każdym gadżecie w polu gg_GadgetID wpisaliśmy inną liczbę.

```
gg_GadgetIDequ 38

                                ; w A2 jest już adres gadżetu,
                                ; który został wciśnięty
    move.w  gg_GadgetID(a2),d0  ; pobranie numeru gadżetu
```

Przedstawione powyżej postępowanie jest opisem działania procedury "CzekajGadget" z listingu kalkulatora i myślę, że jego zrozumienie nie sprawi nikomu większych problemów.

W programie pojawiły się też nowe - nie opisywane dotychczas - funkcje.

Jedną z nich jest funkcja DrawBorder z biblioteki Intuition. Funkcja ta rysuje na ekranie ramkę zdefiniowaną przez strukturę Border (taką samą jak opisywana w tym artykule ramka wokół gadżetu). DrawBorder potrzebuje w rejestrze A0 adresu RastPortu okna, na którym chcemy narysować ramkę, w A1 adresu struktury Border, a w D0 i D1 odpowiednio współrzędnych X i Y lewego, górnego rogu rysowanej ramki (do współrzędnych tych dodane zostaną jeszcze współrzędne zawarte w definicji linii w strukturze Border). Przykładowe użycie funkcji DrawBorder może wyglądać tak:

```
LVODrawBorder equ -108
wd_RPort equ 50

    move.l  #Ramka,a1           ; adres definicji jakiejś ramki
    moveq#0,d0                  ; współrzędna X
    moveq#0,d1                  ; współrzędna Y
    move.l  WindowBase(pc),a0   ; adres struktury Window
    move.l  wd_RPort(a0),a0     ; adres struktury RastPort
                                ; dołączonej do tego okna
    move.l  IntuiBase(pc),a6    ; baza biblioteki Intuition
    jsr  _LVODrawBorder(a6)     ; wywołanie funkcji DrawBorder
```

W naszym programie funkcja DrawBorder została użyta w procedurze "RysujWyswietlacz", która rysuje prostokąt wewnątrz którego pojawiają się cyfry (tak jak na prawdziwym kalkulatorze).

Drugą z nowych funkcji jest PrintIText - również z biblioteki Intuition. Funkcja ta służy do wyświetlenia tekstu zdefiniowanego za pomocą struktury IntuiText. Przed wywołaniem PrintIText należy w rejestrze A0 umieścić adres RastPortu związanego z okienkiem, na który chcemy wydrukować tekst, w A1 - adres struktury IntuiText, natomiast w rejestrach danych D0 i D1 współrzędnych X i Y. Współrzędne te zostaną dodane do współrzędnych zawartych w IntuiText (it_LeftEdge i it_TopEdge) i dopiero w obliczonej w ten sposób pozycji zostanie wydrukowany tekst. Oto przykład użycia funkcji PrintIText:

```
LVOPrintIText equ -216
wd_RPort  equ 50

    moveq#0,d0                  ; współrzędna X tekstu
    moveq#0,d1                  ; współrzędna Y tekstu
    move.l  WindowBase(pc),a0   ; adres struktury Window
    move.l  wd_RPort(a0),a0     ; adres struktury RastPort
    move.l  #TEKST,a1           ; adres definicji tekstu do wyświetlenia
    move.l  IntuiBase(pc),a6    ; baza biblioteki intuition
    jsr  _LVOPrintIText(a6)     ; wyświetlenie tekstu
```

W naszym programie PrintIText służy do wyświetlania zawartości wyświetlacza naszego kalkulatora (procedura "WyswietlaczPisz").

Po wpisaniu i uruchomieniu programu z listingu zauważycie zapewne, że na razie kalkulator posiada jedynie klawisze z cyframi i nie wykonuje żadnych działań poza wyświetleniem ostatnio wciśniętej cyfry. I tak być powinno, gdyż dołączeniem klawiszy służących do wybierania działań matematycznych i samymi procedurami wykonującymi te działania zajmiemy się w następnym odcinku kursu asemblera. ❑

```
;    ## Listing kalkulatora 1 ##

; ## procedury exec.library ##
_LVOOpenLibrary equ -552
_LVOCloseLibrary equ -414

_LVOWaitPort equ -384
_LVOGetMsg equ -372
_LVOReplyMsg equ -378

; ## procedury intuition.library ##
_LVOOpenWindow equ -204
_LVOCloseWindow equ -72
_LVOWbenchToFront equ -342
_LVOWbenchToBack equ -336
_LVODrawBorder equ -108
_LVOPrintIText equ -216

wd_RPort equ 50
wd_UserPort equ 86

WINDOWDRAGequ 2             ; okno można przesuwać po ekranie
WINDOWCLOSE equ 8           ; okienko będzie miało gadżet
                            ; służący do jego zamykania
GIMMEZEROZERO equ 1024      ; współrzędna (0,0)
                  ; nie będzie w lewym górnym rogu okna,
                  ; ale wlewym górnym rogu jego części
                  ; roboczej (wewnętrznej)
WBENCHSCREENequ 1           ; okno na ekranie workbencha
GADGHCOMP equ 0             ; gadżet zmienia kolor
                            ; po wciśnięciu
GADGIMMEDIATE equ 2         ; gadżet wysyła komunikat do
                            ; zaraz po wciśnięciu IDCMP
RELVERIFY equ 1             ; po puszczeniu przycisku myszy
                            ; gadżet przejdzie w stan "wyciśnięty"
BOOLGADGET equ 1            ; gadżet typu wciśnięty/wyciśnięty

CLOSEWINDOW equ 512         ; system IDCMP będzie reagował
                  ; na wciśnięcie gadżetu zamykającego okienko
GADGETDOWNequ 32            ; system IDCMP będzie reagował
                  ; na wciśnięcie gadżetów

gg_GadgetIDequ 38

im_Class equ 20
im_IAddress equ 28

; ## procedury graphics.library ##
_LVOMove equ -240
_LVORectFill equ -306
_LVOSetAPen equ -342

    section Program,code    ; poinformowanie assemblera, że
                            ; teraz będzie program, a nie
                            ; dane (bez tego też będzie dobrze)

Start:   jmp Main           ; skok do procedury głównej

; ## Procedury otwarcia i zamknięcia bibliotek oraz okna ##

OpenGraphics:
    move.l 4.W,a6                ; baza biblioteki Exec
    lea GraphicsName(pc),a1      ; nazwa biblioteki Graphics
    moveq#0,d0                   ; nr wersji biblioteki
    jsr _LVOOpenLibrary(a6)      ; otworzenie biblioteki
    move.l d0,GraphBase          ; zapamiętanie adresu bazowego
                                 ; biblioteki Graphics
    rts

OpenIntuition:
    move.l 4.W,a6                ; baza biblioteki Exec
    lea IntuitionName(pc),a1     ; nazwa biblioteki Intuition
    moveq#0,d0                   ; nr wersji biblioteki
    jsr _LVOOpenLibrary(a6)      ; otworzenie biblioteki
    move.l d0,IntuiBase          ; zapamiętanie adresu bazowego
                                 ; biblioteki Intuition
    rts

CloseGraphics:
    move.l 4.W,a6                ; baza biblioteki Exec
    move.l GraphBase(pc),a1      ; baza biblioteki Graphics
    jsr _LVOCloseLibrary(a6)     ; zamknięcie biblioteki
    rts

CloseIntuition:
    move.l 4.W,a6                ; baza biblioteki Exec
    move.l IntuiBase(pc),a1      ; baza biblioteki Intuition
    jsr _LVOCloseLibrary(a6)     ; zamknięcie biblioteki
    rts

OpenWindow:
    move.l IntuiBase(pc),a6      ; baza biblioteki Intuition
    lea WindowDef(pc),a0         ; adres definicji okna
    jsr _LVOOpenWindow(a6)       ; otworzenie okna
    move.l d0,WindowBase         ; zapamiętanie adresu struktury Window
    rts

CloseWdw:
    move.l IntuiBase(pc),a6      ; baza biblioteki Intuition
    move.l WindowBase(pc),a0     ; adres struktury Window
    jsr _LVOCloseWindow(a6)      ; zamknięcie okna
    rts

WindowDef: dc.w 200              ; współrzędna x okna
    dc.w 20                      ; współrzędna y okna
    dc.w 150                     ; szerokość okna
    dc.w 100                     ; wysokość okna
    dc.b0                        ; kolor detali w oknie
    dc.b1                        ; kolor wypełnień w oknie
    dc.l CLOSEWINDOW+GADGETDOWN
                                 ; znaczniki IDCMP
    dc.l WINDOW CLOSE+WINDOWDRAG
                                 +GIMMEZEROZERO
                   ; znaczniki określające wygląd i typ okna
    dc.l Klawisz0GAD   ; adres pierwszej struktury gadget dla tego okna
    dc.l 0
    dc.l WindowName    ; adres tekstu będącego nazwą okna
    dc.l 0             ; wskaźnik struktury screen dla ekranu, na którym
                       ; ma się otworzyć okno (w tym przypadku równy
                       ; zeru bo używamy ekranu workbencha)
    dc.l 0             ; adres struktury BitMap jeśli chcemy używać
                       ; własnych bitmap (my nie chcemy)
    dc.w 0             ; minimalna szerokość okna
    dc.w 0             ; minimalna wysokość okna
    dc.w 150           ; maksymalna szerokość okna
    dc.w 100           ; maksymalna wysokość okna
    dc.w WBENCHSCREEN        ; typ okna (w naszym przypadku okno
                             ; na ekranie workbencha)

WindowName: dc.b 'Kalkulatorek',0        ; nazwa okna

GraphicsName:dc.b 'graphics.library',0   ; nazwa biblioteki Graphics
IntuitionName: dc.b 'intuition.library',0 ; nazwa biblioteki Intuition

GraphBase: dc.l 0        ; miejsce przechowywania bazy Graphics.library
IntuiBase: dc.l 0        ; miejsce przechowywania bazy Intuition.library
WindowBase: dc.l 0       ; miejsce przechowywania adresu
                         ; struktury Window dla okienka

WBtoFront:
    move.l IntuiBase(pc),a6
    jsr _LVOWbenchToFront(a6)
    rts

WBtoBack:
    move.l IntuiBase(pc),a6
    jsr _LVOWbenchToBack(a6)
```

```
    rts

;   ## Program główny ##
Main:
    bsr  OpenGraphics       ; wywołanie procedury
                            ; otwierającej bibliotekę Graphics
    beq  NoGraphics         ; skok jeśli błąd otwarcia Graphics
    bsr  OpenIntuition      ; wywołanie procedury
                            ; otwierającej bibliotekę Intuition
    beq  NoIntuition        ; skok jeśli błąd otwarcia Intuition
    bsr  WBtoFront          ; ustawienie ekranu workbencha
                            ; przed innymi ekranami
    bsr  OpenWindow         ; wywołanie procedury otwierającej okno
    beq  NoWindow           ; skok gdy błąd otwarcia okna

    bsr  Kalkulator         ; wywołanie głównego podprogramu

    bsr  CloseWdw           ; wywołanie procedury
NoWindow:                   ; zamykającej okno
    bsr  WBtoBack           ; ustawienie ekranu workbencha
                            ; za innymi ekranami
    bsr  CloseIntuition     ; wywołanie procedury zamykającej
NoIntuition:                ; bibliotekę Intuition
    bsr  CloseGraphics      ; wywołanie procedury zamykającej
NoGraphics:                 ; bibliotekę Graphics
    rts                     ; powrót z programu

;  ## Definicje gadżetów klawiszy kalkulatora ##

Klawisz0GAD: dc.l Klawisz1GAD   ; adres następnego gadżetu
    dc.w  5                     ; x górnego, lewego rogu
    dc.w  70                    ; y górnego, lewego rogu
    dc.w  25                    ; szerokość
    dc.w  13                    ; wysokość
    dc.w  GADGHCOMP             ; po wciśnięciu gadżetu będzie
                                ; on zmieniał kolor
    dc.w  GADGIMMEDIATE+RELVERIFY ; gadżet będzie
                                ; reagował i na wciśnięcie i na zwolnienie
    dc.w  BOOLGADGET            ; gadżet typu wciśnięty/wyciśnięty
    dc.l  Ramka1,0              ; adres definicji ramki gadżetu
    dc.l  Tekst0                ; adres definicji tekstu w polu gadżetu
    dc.l  0,0                   ; MutualExclude i SpecialInfo
                                ; (nie będziemy tego używać)
    dc.w  0                     ; numer gadżetu
    dc.l  0                     ; nie będziemy tego wykorzystywać

Klawisz1GAD: dc.l Klawisz2GAD
    dc.w  5,55
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst1
    dc.l  0,0
    dc.w  1
    dc.l  0
Klawisz2GAD: dc.l Klawisz3GAD
    dc.w  35,55
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst2
    dc.l  0,0
    dc.w  2
    dc.l  0
Klawisz3GAD: dc.l Klawisz4GAD
    dc.w  65,55
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst3
    dc.l  0,0
    dc.w  3
    dc.l  0
Klawisz4GAD: dc.l Klawisz5GAD
    dc.w  5,40
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst4
    dc.l  0,0
    dc.w  4
    dc.l  0
Klawisz5GAD: dc.l Klawisz6GAD
    dc.w  35,40
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst5
    dc.l  0,0
    dc.w  5
    dc.l  0
Klawisz6GAD: dc.l Klawisz7GAD
    dc.w  65,40
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst6
    dc.l  0,0
    dc.w  6
    dc.l  0
Klawisz7GAD: dc.l Klawisz8GAD
    dc.w  5,25
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst7
    dc.l  0,0
    dc.w  7
    dc.l  0
Klawisz8GAD: dc.l Klawisz9GAD
    dc.w  35,25
    dc.w  25,13
    dc.w  GADGHCOMP,GADGIMMEDIATE
                    +RELVERIFY,BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst8
    dc.l  0,0
    dc.w  8
    dc.l  0
Klawisz9GAD: dc.l 0          ; 0 jeśli nie chcemy już więcej gadżetów
    dc.w  65,25
    dc.w  25,13
    dc.w  GADGHCOMP
    dc.w  GADGIMMEDIATE+RELVERIFY
    dc.w  BOOLGADGET
    dc.l  Ramka1,0
    dc.l  Tekst9
    dc.l  0,0
    dc.w  9
    dc.l  0

Ramka1:   dc.w  0,0          ; x,y górnego,lewego rogu
    dc.b  1,0                ; kolor linii
    dc.b  1                  ; tryb rysowania (1 to zwykłe
                             ; rysowanie RP_JAM2)
    dc.b  5                  ; ile linii
    dc.l  LineTab1           ; adres tabeli współrzędnych punktów
    dc.l  0                  ; koniec definicji ramki

LineTab1: dc.w  0,0          ; x,y pierwszego punktu
    dc.w  24,0               ; x,y drugiego punktu
    dc.w  24,12              ; x,y trzeciego punktu
    dc.w  0,12               ; x,y czwartego punktu
    dc.w  0,0                ; x,y piątego punktu

Tekst0:   dc.b 1             ; kolor tekstu
    dc.b  0                  ; kolor tła
    dc.b  1,0                ; tryb rysowania
    dc.w  8,3                ; x,y górnego,lewego rogu
    dc.l  0                  ; adres struktury definiującej
                             ; rodzaj czcionki (gdy 0, to
                             ; używany jest standardowy font)
```

```
    dc.l Txt0                  ; adres właściwego tekstu
    dc.l 0                     ; koniec definicji tekstu gadżetu
Tekst1:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt1,0
Tekst2:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt2,0
Tekst3:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt3,0
Tekst4:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt4,0
Tekst5:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt5,0
Tekst6:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt6,0
Tekst7:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt7,0
Tekst8:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt8,0
Tekst9:    dc.b 1,0
    dc.w  0,8,3
    dc.l 0,Txt9,0

Txt0:    dc.b '0',0            ; teksty muszą być zakończone zerem
Txt1:    dc.b '1',0
Txt2:    dc.b '2',0
Txt3:    dc.b '3',0
Txt4:    dc.b '4',0
Txt5:    dc.b '5',0
Txt6:    dc.b '6',0
Txt7:    dc.b '7',0
Txt8:    dc.b '8',0
Txt9:    dc.b '9',0

; ## Procedury kalkulatora ##

Kalkulator:
    bsr  RysujWyswietlacz          ; narysowanie ramki wyswietlacza
Kalkulator1:
    bsr  CzekajGadget
    cmpi.b  #255,d0
    beq KalkulatorEND
    addi.b #'0',d0
    move.b  d0,WyswietlaczTXT+7
    bsr  WyswietlaczPisz           ; wypisanie stanu wyswietlacza
    bra  Kalkulator1
KalkulatorEND:
    rts


RysujWyswietlacz:
    move.l  #WyswietlaczRamka,a1
    moveq#0,d0
    moveq#0,d1
    move.l  WindowBase(pc),a0      ; adres struktury Window
    move.l  wd_RPort(a0),a0        ; adres struktury RastPort
                                   ; dołączonej do naszego okna
    move.l  IntuiBase(pc),a6       ; baza biblioteki Intuition
    jsr  _LVODrawBorder(a6)        ; narysowanie ramki
    rts

WyswietlaczRamka:
    dc.w  5,5
    dc.b 1,0
    dc.b0
    dc.b5
    dc.l WyswietlaczLinie
    dc.l 0
WyswietlaczLinie:
    dc.w  0,0
    dc.w  130,0
    dc.w  130,15
    dc.w  0,15
    dc.w  0,0

CzekajGadget:
    move.l  WindowBase(pc),a0      ; adres struktury Window
    move.l  wd_UserPort(a0),a0     ; adres struktury UserPort
                                   ; dołączonej do naszego okna
    move.l  4.W,a6                 ; baza biblioteki Exec
    jsr  _LVOWaitPort(a6)          ; czekanie na sygnał od IDCMP

    move.l  WindowBase(pc),a0
    move.l  wd_UserPort(a0),a0
    move.l  4.W,a6
    jsr  _LVOGetMsg(a6)            ; pobranie komunikatu od IDCMP
    tst.l d0                       ; sprawdzenie rejestru D0
    beq CzekajGadget               ; jednak nie ma komunikatu

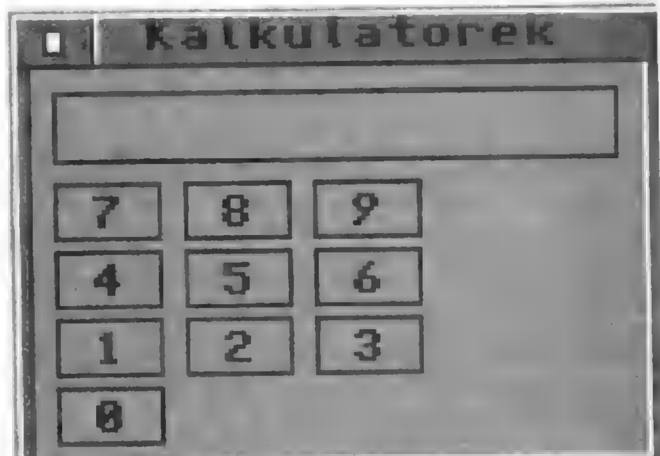
    move.l  d0,a1
    move.l  im_Class(a1),d2        ; pobranie typu komunikatu
    move.l  im_IAddress(a1),a2     ; pobranie adresu gadżetu,
                                   ; który wysłał komunikat
    move.l  4.W,a6                 ; baza biblioteki Exec
    jsr  _LVOReplyMsg(a6)          ; potwierdzenie odebrania

    cmpi.l #CLOSEWINDOW,d2
    beq ZamkniecieOkna
    move.w  gg_GadgetID(a2),d0     ; pobranie numeru gadżetu
    rts
ZamkniecieOkna move.w  #255,d0     ; 255 będzie oznaczało wciśnięcie
                                   ; gadżetu zamknięcia okienka
    rts

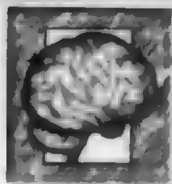
WyswietlaczPisz:
    moveq#50,d0                    ; współrzędna X tekstu
    moveq#9,d1                     ; współrzędna Y tekstu
    move.l  WindowBase(pc),a0
    move.l  wd_RPort(a0),a0
    move.l  #WyswietlaczTEKST,a1   ; adres definicji tekstu do wyswietlenia
    move.l  IntuiBase(pc),a6       ; baza biblioteki Intuition
    jsr  _LVOPrintIText(a6)        ; wyswietlenie zawartosci
    rts                            ; wyświetlacza

WyswietlaczTEKST:
    dc.b 1                         ; kolor tekstu
    dc.b0                          ; kolor tła
    dc.b 1,0                       ; tryb rysowania
    dc.w  0,0                      ; x,y górnego,lewego rogu
    dc.l 0                         ; adres struktury definiującej rodzaj
                                   ; czcionki (gdy 0, to
                                   ; używany jest standardowy font!)
    dc.l WyswietlaczTXT            ; adres właściwego tekstu
    dc.l 0                         ; koniec definicji tekstu
WyswietlaczTXT:
    dc.b'        ',0               ; (między pierwszym
                                   ; apostrofem, a kropką jest osiem spacji)
```

Uwaga! Tekst wytłuszczony należy wpisywać w jednej linii!



Wygląd okienka uzyskanego po uruchomieniu programu z listingu.



RUSZ GŁOWĄ

n a i f t a u j a m i g a x e x a k m l f d b q n a d e s z ł a
y p c o b c v z y l h q n p d v r o a p k s v a w x t g z u z r
g r l s h f d o e z d t o o x w o u a x e c g h i u f g w k d o
ć a w o ż a g n a a z t d j a ł p f m b i ż y t t g l h r e h p
m q w c l r r y i b w y t r b u u r c k i z y h n q k c p r k u
r s p r a w a m u v z a w k l k m b c g f m d x p r z y u d r m
e r g i m q f t e q n n a w i ł g q f a k t z u i w k z h n ł g
i n ó v o p i c o b t c r i t f j j e b p y f b f e z s w a z j
c v l c f l k s w d w z z d t x k p s j q m h u e s d j k e v k
o i r o x ę i s w o h c a y e f j r i n g m h y f i j e q z h j
p l d m e f p s ł t h i n o r o g o t y u h c y n z c i f a r g
o k h ł a q m b z r n q l z o r u s t a z b m g h m l n p a q u
ł v n s n c l c i a s r u q c y t t h ł f c ś z a b x ż h k p t
k o m p u t e r o w e j i o p e t a c j a c h t x q z a n m o t
w m o x l p g n e p j q s v o y ł ł u o ń q e j a d y w r x s ł
ł s d p d p y e l b z a a q a i y f w e a ł e m p c t j v u i y
w i e v n r m i e j s c u t n j n m r ó w n y h w g n a t h a n
l w m k j f g n m v b g s p c p v ą x j o w i i b o q n f z d v
f w r o p y o z c t l a n w b g a j u y s r d s m ę h k k x a a
z d y p v ę g a w u ł e r o ż k l a e n o q ł f j i o c m w o l
r j l r f d j j v k a d w a i v n g u d t d ł x y s m q a h d n
p o x o p t r a o l m x b q f l u a b x s x u n b h n s g p z u
r w z c r k c h t d q o r ó m g k m e z a r b g i l y e r ó t k
ł w e e o e t n k j e d n y m s p o a s z s x s d d r m j e b p
z w h s u ł a y f o t p m v a q m p j l o c l u i w o m e d o m
w j p o ł b t u w i k y z u m i k d k u k a v w v a r s f u o k
r t y r h b l e p w q y u t y s y v g b h o g j l t q o c x j y
ó h y ó ł c o g y b a t c m z ó v p o b r a z u ł s i g g x e v
c x j w j z ł v w i k n b ł b s t g r h t l a p v e n y f i h t
i v o r w y s t a r c z y a u w x m a i m y j t p z w l n b l x
ć l f t ł l s ł h v p i p z a q ł o r f h p ą j c a m i n a j ł
j s p e c i a l i z o w a n y c h n l e h j ć p s o t v h a x h

Witam wszystkich i zapraszam do kolejnego "ruszania głową". Na początku czas na rozwiązanie zagadki z Amigowca 1/94. Poprawna odpowiedź brzmi:

Marek,	A600,	Gdańsk,	Dema,
Wojtek,	A3000,	Bydgoszcz,	DTP,
Krzysiek,	A1000,	Wrocław,	Grafika,
Michał,	A500,	Poznań,	Gry,
Robert,	A4000/030,	Kraków,	Video.

Nagrody w postaci gier wylosowali:
Tomasz Kos - Ruda Śląska 10
Agnieszka Chruściel - Łódź
Piotr Andrychiewicz - Końskie
Nagrody prześlemy pocztą.

Tym razem kolej na nieco odmienne zadanie, które przeznaczam szczególnie dla naszych stałych czytelników. Do jego rozwiązania nie jest potrzebny duży zasób wiedzy z informatyki, natomiast przyda się odrobina spostrzegawczości i ... poprzedni numer Amigowca. W kwadracie składającym się z 1024 liter znajdują się ukryte wyrazy, które mogą być wypisane pionowo i poziomo, zarówno wprzód, jak i wspak. Poza tym mogą się stykać i krzyżować. Po wyszukaniu wszystkich wyrazów, należy wziąć poprzedni numer Amigowca i wśród artykułów znaleźć takie cztery zdania, które składają się ze znalezionych wyrazów. Dla ułatwienia podaję, że nie trzeba ponownie czytać całych artykułów. Wszystkie zdania znajdują się możliwie blisko ich początków.

Odnalezione zdania należy przepisać na kupon konkursowy i przesłać na adres redakcji:

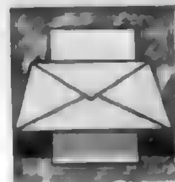
"Amigowiec"
dział Rusz Głową
ul. Świętojańska 2/7
85-017 Bydgoszcz

Wśród poprawnych odpowiedzi zostaną rozlosowane cenne nagrody.

KUPON KONKURSOWY nr 9
Szukane zdania

1. ......
2. ......
3. ......
4. ......

Imię nazwisko: ......
Adres: ......
Wiek (nie trzeba wypełnić): ......
Zawód (nie trzeba wypełniać): ......



Czy można połączyć ze sobą A500+ [illegible] A4000 i nakłonić je do współpracy?

Czy będzie potrzebne tworzenie sieci (konieczny zakup kart sieciowych), czy też wystarczy połączenie przez np. złącze równoległe, szeregowe lub też SCSI (między kontrolerem Protar, [illegible] kartą SCSI w A4000)?

Co to jest kabel Null Modem?

Czy 3 MB RAMu pochodzące z rozszerzenia pamięci A500+ dadzą się podłączyć we wspólny blok ciągłej pamięci w A4000 (bez przenoszenia kości, czyli po bezpośrednim połączeniu komputerów)?

Wspomniane w pierwszym pytaniu komputery, powinno dać się połączyć ze sobą bez większych problemów. W najprostszym przypadku nie są wymagane dodatkowe elementy, w postaci kart sieciowych. Po zaopatrzeniu się w odpowiedni "kabelek" (to właśnie ów Null Modem z trzeciego pytania) można połączyć bezpośrednio A500+ i A4000, wykorzystując do tego celu port szeregowy. Po zastosowaniu odpowiedniego oprogramowania np. ParNet jest możliwe wykonywanie funkcji sieci peer-to-peer (komputery sieci są jednostkami równorzędnymi).

Wspomnianych 3 MB pamięci RAM z A500+ nie da się w żaden sensowny sposób wykorzystać w Amidze 4000 (pomijając fakt, że jest to pamięć 16-bitowa).

Nie spotkałem się jeszcze [illegible] połączeniem komputerów przez złącze SCSI. Zazwyczaj jest ono wykorzystywane do podłączania różnego rodzaju urządzeń zewnętrznych (np. SyQuest lub CD-Rom), nie zaś do łączenia komputerów w celu wymiany danych czy zastosowań sieciowych (choć taka możliwość istnieje po zakupieniu odpowiedniego oprogramowania i kontrolerów).

Jaki byłby aktualnie koszt zakupu A4000 [illegible] pełnej wersji (a więc [illegible] koprocesorem i MMU)?

Amiga 4000/040 (czyli wersja z procesorem 68040 oraz zintergrowanym z nim koprocesorem i układem zarządzającym pamięcią - MMU) w minimalnym zestawie kosztuje ok. 4000 DM. W skład takiego zestawu wchodzi dysk twardy o pojemności 120 MB oraz standardowo 6 MB RAM. Cena zależy oczywiście od konkretnej konfiguracji.

Ile kosztuje rozszerzenie A4000 kontroler SCSI oraz jakie są [illegible] dołożenia do niej wyjścia Composite Video?

Czy modulator [illegible] A500+ będzie poprawnie pracował [illegible] wyjściu RGB A4000?

Rozszerzenie A4000 o kontroler SCSI to wydatek rzędu 250 DM (bez HDD). Jeżeli chodzi o kontroler SCSI II, który jest o wiele wydajniejszy w pracy z A4000 to koszt waha się w granicach 700 DM. Pozostałe dwa pytania są bezpośrednio ze sobą połączone. Otóż, w złącze Composite Video można zaopatrzyć się kupując kartę graficzną (złącze takie stanowi jej standardowe (DCTV) lub dodatkowe (Retina, Picollo) wyposażenie) lub stosując modulator telewizyjny (np. taki od A500). W przypadku stosowania modulatora musimy liczyć się z tym, iż nie będzie można korzystać z takich trybów jak DblPAL i DblNTSC, gdyż pracują one z częstotliwościami odchylania poziomego wyższymi niż palowskie 15,6 kHz (lub dla NTSC 15,72 kHz).

W jaki sposób zdobyć opis wyjścia SCSI, znajdującego się [illegible] kontrolerze firmy "Protar"?

Złącze SCSI jest standardem światowym i jego budowa jest w każdym przypadku jednakowa. O ile nie zostało ono opisane w oryginalnej instrukcji (a powinno), proponowałbym poszukać dowolnej instrukcji obsługi do komputera, bądź urządzenia, które takie złącze posiada.

Czy do Amigi CDTV da się zamontować kości AGA?

Niestety, taki bezpośredni zabieg wymiany kości graficznych nie może zakończyć się powodzeniem. U podłoża tego faktu leżą zbyt duże różnice w wewnętrznej budowie obu komputerów. Pojawiły się ostatnio informacje (niesprawdzone) mówiące o specjalnych zestawach pozwalających przebudować A500 na A1200 (jak oni to zrobili?!), lecz nigdzie nie znalazłem wzmianek o analogicznej możliwości dla Amigi CDTV.

Czy programem True Kick v2.0 można zrobić emulator (programowy) Amigi 500 (wyłączyć kompakt itd.), czy należy dokonać jakiejś przeróbki?

Nie można zrobić tego typu emulatora. Jedyną naprawdę efektywną "emulację" A500 na Amiga CDTV zapewnia specjalna, sprzętowa przeróbka. Przeróbkę taką oferowała do niedawna firma Elsat tel. 40-58-76 (Warszawa). Tematyka ta pojawiała się już na łamach "Pism, pisemek" w numerze 2/94, tak więc zainteresowanych tym problemem odsyłam do poprzednich numerów "Amigowca".

Czy przy PC-Tascu można korzystać z dyskietek HD (na Amiga CDTV), czy należy kupić stację dysków HD?

Standardowa stacja dysków dołączana do zestawu z Amiga CDTV nie pozwala na korzystanie z dysków HD. Jednak po zaopatrzeniu się w odpowiedni kontroler i napęd nic nie stoi na przeszkodzie, aby w PC-Tascu stosować dyskietki o podwójnej gęstości zapisu. Sam program umożliwia pracę z dyskietkami HD.

Czy chcąc wgrać program [illegible] "Spectruma" wystarczy podłączyć magnetofon przez sampler?

Do wgrania programu pochodzącego z mikrokomputera "Spectrum" rzeczywiście wystarcza podłączenie magnetofonu przez sampler. Oczywiście należy wcześniej zaopatrzyć się w odpowiedni emulator tego "komputerka". Jednak poza względami sentymentalnymi nie widzę specjalnego zastosowania dla tego typu zabaw.

Czy 1 MB Chip odpowiada 1 MB Fast RAM?

Zagadnienie różnic pomiędzy pamięcią Chip a pamięcią Fast ciągle powtarza się w listach przychodzących do redakcji. Pomimo, iż problem ten był już niejednokrotnie omawiany, postaram się raz jeszcze przedstawić te dwa rodzaje pamięci.

Chip RAM to pamięć bezpośrednio dostępna dla wyspecjalizowanych układów Amigi. Mogą być w niej przechowywane dane dotyczące grafiki, dźwięku, bufora stacji dysków, dysku twardego, itd. Ilość dostępnej pamięci tego rodzaju, zależy od wersji układu Agnus. Szybki rozwój komputerów Commodore Amiga spowodował stopniową ewolucję tego specjalizowanego chipa. Od "zwykłego" Agnusa (A500 z Kickstartem 1.2), przez Fat Agnus (Kickstart 1.3), Super Fat Agnus (A500+, A600, A2000D, A3000) po zupełnie nową kość Alice, montowaną w A1200 i A4000. Specyfika wewnętrznej budowy Amigi powoduje, że nawet duża ilość pamięci innego typu niż Chip nie gwarantuje poprawnego funkcjonowania programów. Związane jest to [illegible] zbyt małym obszarem ciągłej pamięci dostępnej dla układów zajmujących się grafiką, dźwiękiem, czy operacjami wejścia-wyjścia.

Rola Fast RAM jest nieco inna. Do pamięci tego rodzaju ma bezpośredni dostęp główny procesor Amigi. Praca z pamięcią Fast RAM jest niezależna od tego, jak bardzo obciążymy kanały DMA (Direct Memory Access - pozwalające na transmisję danych z dyskietek, danych wyświetlanego obrazu, duszków, itp.). Dzięki takiemu potraktowaniu sprawy nasza Amiga może przyspieszyć o ok. 30 do 35% (dla programu wykonywanego w pamięci Fast RAM). Jednak należy pa



PISMA, PISEMKA

miętać, że wszystko co uzyskujemy na ekranie lub membranie głośnika, przechodzi przez Chip RAM.

Jak widzimy 1 MB Chip RAM nie odpowiada 1 MB Fast RAM, gdyż rola jaką spełniają te dwa rodzaje pamięci jest diametralnie różna.

Jaki kontroler dysku twardego wybrać i jakiej pojemności (zajmuję się głównie obróbką tekstu, nieskomplikowaną grafiką i grami - rzadko), jeżeli posiadam A500 z 1 MB RAM?

Jeszcze niedawno rozbudowa A500 o dysk twardy była sprawą niezwykle kosztowną. Jednak pojawienie się nowych modeli Amig i pewnego rodzaju "nasycenie" sprzętem komputerowym spowodowało radykalny spadek cen (niestety dotyczy to głównie sprzętu używanego). Jeżeli autor pytania rzeczywiście chce ograniczyć się do zastosowań wymienionych w pytaniu, to jego potrzeby w zupełności zaspokoi kontroler klasy AT-bus z napędem dysków twardych o pojemności już od 40 MB. Taki zestaw, w przypadku zakupu sprzętu używanego, nie powinien przekroczyć 2,5 mln zł. Nowy kontroler At-bus kosztuje ok. 1,7 mln zł, a napęd HDD 40 MB, w zależności od producenta, to wydatek rzędu 1,6-2,5 mln zł (czyli sumarycznie cena nowego sprzętu wyniesie ok. 3,5-4 mln zł). W tym miejscu pojawia się problem nieco innego rodzaju. Ilość pamięci RAM w postaci 1 MB przestanie być wystarczająca. Po wczytaniu systemu i doliczeniu bufora dysku twardego, pozostanie raptem ok. 800 KB wolnej pamięci. Dla większości programów jest to naprawdę minimalna ilość RAM, a może się zdarzyć, że niejedna gra odmówi pracy. Optymalnym rozwiązaniem byłby zakup kontrolera z możliwością rozbudowy pamięci. Zestaw umożliwiający dodatkową instalację pamięci RAM, w postaci osobno dodawanych kości SIMM, kosztuje ok. 2.7 mln zł (np. "Mega RAM HD" - firmy Elsat, cena oczywiście bez HDD).

Gdzie można znaleźć procedury Font Convert.amos i Keyboard Definer.amos potrzebne [illegible] wprowadzenia własnego zestawu znaków i klawiatury do edytora Amosa V1.3?

Procedury te wchodzą w skład pakietu AMOS THE CREATOR od samego początku, czyli pierwszych wersji tego języka programowania. Jeśli więc (tak jak pisze autor pytania) nie znajdują się one na posiadanych przez niego dyskach, to przypuszczam, że jest to piracka i niekompletna kopia oryginalnego pakietu. Niestety, często tak się zdarza, iż w programach pochodzących z giełdy brakuje pewnych, mniej znaczących (w mniemaniu handlarza) fragmentów. Jak do tej pory jest to w naszym kraju zjawisko co najmniej tak powszechne jak samo piractwo. AMOS jest programem komercyjnym i wspomniane w pytaniu procedury stanowiące jego część, nie mogły znaleźć się na wydawanych przez nasze czasopismo dyskach PD. W tej sytuacji mogę poradzić jedynie zakup oryginalnego pakietu AMOSa lub zainteresowanie się nowym produktem jakim jest AMOS PROFESSIONAL, którego edytor przejmuje definicję zestawu znaków i klawiatury z systemu. Zatem nie ma problemów z ich zmianą. Ponadto prowadzony na łamach AMIGOWCA kurs programowania od odcinka nr 11 będzie oparty właśnie na tym pakiecie.

Ostatnio otrzymaliśmy sporą ilość listów z pytaniami dotyczącymi programowania w assemblerze. Postaramy się tutaj dać odpowiedź na te z nich, na które można odpowiedzieć stosunkowo krótko. Na pozostałe będziemy udzielać odpowiedzi w odrębnych artykułach na łamach naszego pisma.

Czy trzeba zamykać biblioteki przy wychodzeniu z programu?

Właściwie nie ma obowiązku zamykania bibliotek przy wychodzeniu z programu, jednak bardziej elegancko jest, gdy je pozamykamy. Każda biblioteka posiada "open count", czyli licznik otwarć, który jest zwiększany o jeden przy każdej próbie otwarcia (nawet już otwartej) biblioteki, a zmniejszany przy próbach zamknięcia. Dobrze by było, gdyby nieużywana biblioteka miała ten licznik wyzerowany. Tak więc, mimo iż zamykanie bibliotek nie jest absolutnie konieczne, lepiej je zamknąć.

Jak pobrać parametry dla programu z okna CLI?

Pobranie parametrów z okienka CLI jest bardzo proste. Każdy program uruchamiany z CLI otrzymuje w rejestrze A0 adres tekstu, który użytkownik napisał za nazwą programu. Tekst ten kończy się wartością 10 (jest to kod zmiany linii). Jednocześnie w rejestrze D0 znajduje się długość tego tekstu (łącznie ze znakiem zmiany linii). Pobranie parametrów polega więc na umiejętnym zanalizowaniu pamięci pod adresem zawartym w rejestrze A0. Przykładowy fragment programu reagującego na opcję -? mógłby wyglądać następująco:

Listing 1

```
Start   cmpi.b #'-',(a0)
                ; czekanie, aż wystąpi minus
    beq JestMinus
    adda.l  #1,a0
                ; ominięcie nieważnego znaku
    subq.w #1,d0
                ; zmniejszenie licznika znaków
    bne Loop1
    bra NieMaParametrow      ; skok gdy
                ; nie ma już znaków w linii CLI
JestMinus  adda.l  #1,a0
    cmpi.b #'?',(a0)     ; sprawdzenie czy
                ; po '-' występuje '?'
    bne Start    ; jeśli nie, to szukamy dalej
Opcja1   ...     ; jeśli tak, to wykonujemy
                 ; program
    rts          ; z tą opcją
NieMaParametrów rts
```

Jaką funkcję pełnią rejestry o adresach $DFF0AC, $DFF0AE, $DFF0BC, $DFF0BE, $DFF0CC, $DFF0CE, $DFF0DC, $DFF0DE ?

Rejstry te nie pełnią ŻADNEJ funkcji. Są po prostu nieużywane i nawet jeśli wpisywanie do nich różnych wartości powoduje jakieś efekty, to NIE NALEŻY ich używać.

Jak się robi błysk całego ekranu, tak jak przy sytuacjach błędnych?

Do błyśnięcia ekranem wykorzystuje się funkcję DisplayBeep z biblioteki Intuition. Funkcji tej należy podać w rejestrze A0 adres struktury Screen połączonej z ekranem, którym chcemy błysnąć. Jeśli chcemy błysnąć ekranem WorkBencha, albo zbytnio nie wiemy, jaki jest adres struktury Screen naszego ekranu, możemy wpisać do A0 wartość 0, [illegible] błyśnięcie będzie obejmowało wszystkie ekrany. ❑

Listing 2

```
_LVOOpenLibrary =-552
_LVOCloseLibrary=-414

_LVODisplayBeep =-96

Flash   move.l 4,a6
    lea IntuitionName(pc),a1
    moveq #0,d0
    jsr  _LVOOpenLibrary(a6)
            ; otwarcie intuition.library
    move.l d0,a6      ; baza intuition do a6
    move.l #0,a0      ; błyskamy wszystkimi
                      ; ekranami
    jsr  _LVODisplayBeep(a6)
                      ; błyśnięcie ekranem
    move.l a6,a1
    move.l 4,a6
    jsr  _LVOCloseLibrary(a6)
            ; zamknięcie intuition.library
    rts
IntuitionName  dc.b  'intuition library',0
```



Public Domain 3/94

Mariusz Ławicki

Kolejny numer Amigowca i oczywiście dyskietka Public Domain. Jak zwykle znalazły się na niej mniej lub bardziej przydatne programy użytkowe, moduł muzyczny oraz wspaniała gra, która będzie odpowiedzialna za Wasze czerwone oczy i nieprzespane noce.

1. GRABIFF

Grabiff służy do wyciągania grafiki z różnych programów. Jest on podobny do prezentowanego już na naszych dyskietkach ScreenXa, lecz ma większe możliwości. Programem tym można zapisywać na dyskietkę całe ekrany lub ich fragmenty, okna oraz wskaźnik myszy. Przed uruchomieniem programu należy go odpowiednio skonfigurować. Można to zrobić dopisując do Tooltypes ikony podane niżej parametry. Jeżeli uruchamiamy program z poziomu CLI należy podać parametry po nazwie programu. Pora teraz na opis wspomnianych parametrów.

On - uaktywnienie programu.

Off - deaktywacja. Deaktywację uzyskujemy również przez ponowne uruchomienie programu.

Quiet - normalnie po uruchomieniu wyświetlany jest requester informujący o gotowości do pracy lub usunięciu programu z pamięci. Ustawienie Quiet spowoduje, że requester się nie pojawi.



Usage - uruchomienie programu powoduje jedynie wyświetlenie okna ze spisem dostępnych parametrów oraz krótkim opisem każdego z nich.

AskName - standardowo pliki z grafiką zapisywane są pod nazwą Grabiff i trzycyfrowym rozszerzeniem. Ustawienie AskName sprawia, że przed zapisem, wyświetlony zostanie requester dyskowy i będziemy mogli podać ścieżkę dostępu oraz nazwę naszego pliku.

CheckOff - opcja wyłącza sprawdzanie, czy plik o podanej nazwie istnieje już na dysku.

CrosshairOff - opcja wyłącza krzyżyk pojawiający się w miejscu kursora myszy po aktywacji programu, nieaktywna jest wtedy możliwość zapisu fragmentu ekranu.

PackOff - normalnie pliki zawierające grafikę są kompresowane. Ustawienie tej opcji wyłącza kompresję.

FrontScreen - standardowo program zapisuje aktualnie aktywny ekran. Ten wariant sprawia, że zapisywany będzie nie ekran aktywny, ale ten, który jest na wierzchu.

Icons - do plików będą dodawane ikony.

Verbase - przy zapisie zostanie wyświetlone okienko zawierające dane obrazka, to znaczy ścieżkę dostępu i nazwę oraz wymiary i ilość bitplanów.

Error=None/Beep/Window - tutaj ustalamy sposób reakcji programu na błędy. Dostępne są trzy możliwości:

None - powoduje ignorowanie błędów,

Beep - reakcją na błędy będzie krótki dźwięk,

Window - parametr standardowy, będą wyświetlane okna.

StandardName - ustawienie standardowej ścieżki dostępu i nazwy zapisywanych plików. Normalnie jest to RAM:Grabiff.XXX, wpisując np. StandardName=DF0:Pic spowodujemy, że pliki będą zapisywane na napędzie DF0: pod nazwami Pic.XXX.

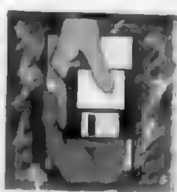
Qualifier=nnn - tutaj ustalamy kombinację klawiszy uaktywniającą program. Wartość nnn jest sumą liczb oznaczających poszczególne klawisze:

Lewy Shift....1
Prawy Shift...2
Caps Lock.....4
Ctrl..........8
Lewy Alt.....16
Prawy Alt....32
Lewa Amiga...64
Prawa Amiga.128

Standardową wartością jest 74, co odpowiada kombinacji Ctrl+Lewa Amiga.

WindowCode=nnn - ustalenie kombinacji, która będzie służyła do zapisu okna. Wartość nnn jest numerem klawisza.

ScreenCode=nnn - jak wyżej, tylko kombinacja służy do zapisu całego ekranu.



PointerCode=nnn - kombinacja do zapisu wskaźnika myszy.

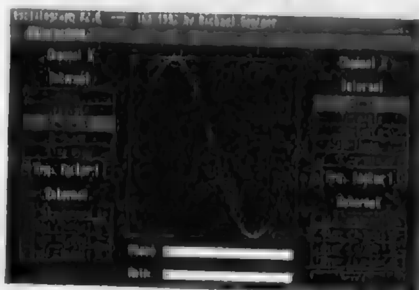
Cx_Priority=nnn - ustawienie priorytetu.

Cx_PopUp=Yes/No - wpisanie Yes spowoduje, że po uruchomieniu programu ukaże się okno, w którym za pomocą gadżetów będziemy mogli ustawić wszystkie wyżej wymienione parametry. Jest to dosyć wygodne rozwiązanie w przypadku częstej zmiany ustawień.

Po uruchomieniu program instaluje się w pamięci i jest gotowy do pracy. Aby go uaktywnić, należy użyć wcześniej zadeklarowanej kombinacji klawiszy. Następnie przytrzymując je, w zależności od tego co chcemy zrobić nacisnąć odpowiedni klawisz. Standardowo jest to "s" dla zapisu całego ekranu, "w" dla zapisu aktywnego okna oraz "p", aby zapisać kursor myszy. Jeżeli chcemy mieć na dyskietce fragment ekranu, to po prostu zaznaczamy go myszką. Do poprawnego działania program potrzebuje pliku GrabiffHandler umieszczonego w katalogu L: oraz co na pewno zmartwi posiadaczy systemu 1.3, wymaga Kickstartu 2.0 lub wyższego. Dużą zaletą programu jest jego konfigurowalność, dlatego polecam Grabiffa wszystkim, którzy lubują się w wyciąganiu grafiki z programów.

Autorem programu jest Bernstein Zirkel.

2. Oscillograph V2.0



Jest to symulator oscyloskopu. Dla tych, którzy nie kochają lutownicy wyjaśniam, że oscyloskop to urządzenie stanowiące skrzyżowanie telewizora z ploterem. Ramiona X i Y plotera zostały zastąpione dwiema parami płytek, między którymi występuje pole elektryczne zmieniające kierunek wiązki elektronów zastępującej pisak. Owa wiązka uderza w luminofor, którym pokryty jest ekran i powoduje jego świecenie. W ten sposób powstaje obraz stanowiący złożenie przebiegów podanych na wejścia X i Y oscyloskopu. Na tej samej zasadzie działa nasz program. Teraz pora na szczegółowy opis jego funkcji i możliwości. Po lewej i prawej stronie ekranu znajdują się identyczne gadżety, przy czym jedne obsługują wejście X, a drugie Y. Za pomocą wspomnianych gadżetów możemy zmienić kształt oraz częstotliwość wprowadzanych na wejścia przebiegów. Standardowymi kształtami są:

Sinus - przebieg sinusoidalny.

Triangle - przebieg trókątny.

Sawtooth - przebieg piłokształtny.

M-Shape - przebieg w kształcie litery M.

Line Mix - przebieg złożony z czterech linii.

Oczywiście kształt każdego z wymienionych przebiegów można dowolnie modyfikować. Wystarczy dwa razy kliknąć na wybrany gadżet, a uruchomi się wbudowany w program edytor. Po lewej stronie ekranu edytora widzimy modyfikowaną falę, natomiast po prawej stronie znajdują się gadżety służące do zmiany jej kształtu:

Name - tutaj podajemy nazwę edytowanego przebiegu.

Load an oscillation from disk - korzystając z tej funkcji możemy załadować zapisane na dyskietce przebiegi. Przykładowe fale znajdują się w katalogu Oscillations.

Save oscillation to disk - zapisanie przebiegu na dyskietce.

Mirror whole oscillation at X - tworzy lustrzane odbicie przebiegu względem osi X.

Mirror whole oscillation at Y - jak wyżej, tylko odniesieniem jest oś Y.

Mirror left half at middle axis - opcja służy do wykonywania przebiegów symetrycznych. Najpierw tworzymy lewą połowę fali, a następnie klikamy na ten gadżet. W efekcie na prawej połowie otrzymamy przebieg symetryczny do wcześniej narysowanego.

Mirror left half at midpoint - działanie tej opcji jest podobne z opisaną wyżej, ale punktem symetrii jest środek "lampy oscyloskopowej".

Function plotter - kreślenie wykresów funkcji matematycznych. Kliknięcie na ten gadżet wywołuje okno, w którym podajemy wzór funkcji oraz ekstremalne wartości X i Y.

Left border - minimalna wartość X.

Right border - maksymalna wartość X.

Bottom border - dolna wartość Y dla układu współrzędnych.

Top border - górna wartość Y dla układu współrzędnych.

Przy obu wyżej wymienionych opcjach znajdują się gadżety **Manual** i **Auto**. Pierwszy z nich pozwala na ręczne wpisanie wartości Y, natomiast, gdy aktywny jest drugi, program sam oblicza maksymalne i minimalne wartości Y.

Function term: f(x)= - pod tym napisem znajduje się okno, w które należy wpisać wzór funkcji jaka ma być wyświetlona. Dostępne są:

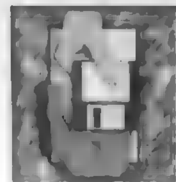
exp(...) - funkcja exponencjalna
log(...) - logarytm o podstawie e
lgd(...) - logarytm o podstawie 10
sqrt(...) - pierwiastek kwadratowy
abs(...) - wartość bezwzględna
sin(...) - sinusoida
cos(...) - cosinusoida
tan(...) - tangensoida
arcsin(...) - odwrotność sinusa
arccos(...) - odwrotność cosinusa
arctan(...) - odwrotność tangensa
sinh(...) - sinus hiperboliczny
cosh(...) - cosinus hiperboliczny
tanh(...) - tangens hiperboliczny
Int(...) - wartość całkowita

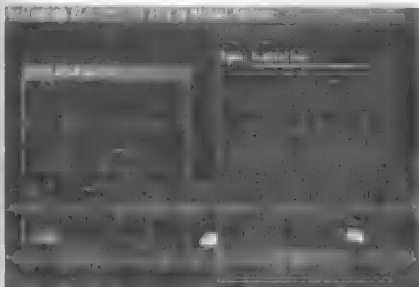
Zdefiniowano również stałą Pi=3.14159

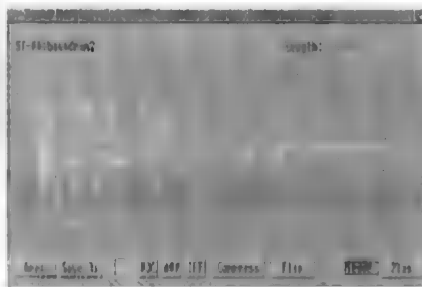
Dla przykładu zróbmy wykres cosinusoidy. Jako graniczne wartości X wpisujemy 0 oraz 2*Pi, jako Y wpisujemy -1 i 1 lub włączamy opcję Auto, następnie jako wzór funkcji podajemy cos(x). Teraz kilkamy na OK i po chwili otrzymamy piękną cosinusoidę. W podobny sposób możemy wykreślić inne funkcje.

Digitizer readout sequence - odczytanie przebiegu z zewnętrznego przetwornika analogowo-cyfrowego.

Undo last action - anulowanie ostatniego polecenia.







Undo all actions - powrót do stanu pierwotnego, przed dokonaniem jakichkolwiek zmian.

Leave editor - wyjście z edytora i powrót do oscyloskopu.

To tyle na temat edytora. Teraz powróćmy do ekranu głównego. Poniżej gadżetów określonych jako Internal, a zawierających kształty fali znajdują się gadżety External, obsługujące zewnętrzne źródła sygnałów. Program może odczytywać zewnętrzne przebiegi za pomocą przetwornika analogowo cyfrowego podłączonego do portu równoległego. Autor przewidział współpracę z czterokanałowym przetwornikiem, którego schemat był publikowany w niemieckim czasopiśmie Amiga Magazin. Można także podłączyć dowolny sampler. Po środku ekranu znajduje się "lampa oscyloskopowa", na której wyświetlany jest wypadkowy przebieg złożony z tego co wprowadzane jest na wejścia X i Y. Poniżej umiejscowione są dwa suwaczki, pierwszym zmieniamy przesunięcie fazowe między sygnałami z wejść X i Y, a drugim włączamy opcję animacji.

Głównym zastosowaniem programu jest demonstracja figur Lissajous, a ze względu na to, że na wejścia oscyloskopu możemy podać praktycznie dowolny przebieg, jest on doskonałym narzędziem do eksperymentów. Można też przeprowadzić pokaz figur Lissajous na lekcji fizyki. Dla elektroników wyposażonych w sampler program będzie mógł zastąpić prosty oscyloskop.

Autorem programu jest Michael Geuter.

3. Sound Machine V1.0

Program ten służy do konwersji plików zawierających dźwięk w postaci sampli. Obsługiwane są cztery podstawowe formaty:

IFF** oraz **RAW - standardowe formaty używane przez Amigę.

WAV - format używany głównie przez programy Microsoft Windows.

VOC - format używany na przykład przez kartę Sound Blaster.

Niestety program nie jest doskonały i nie potrafi konwertować sampli 16-bitowych oraz stereofonicznych zapisanych w formatach WAV i VOC. Obsługa Sound Machine jest bardzo prosta i sprowadza się do klikania na odpowiednie gadżety. Oto ich opis:

Open - ładowanie sampli. Program automatycznie rozpozna format zapisu ładowanej próbki i jeżeli jest to niemożliwe, użyje formatu RAW. Po wczytaniu do pamięci na ekranie zostaje wyświetlony wykres sampla.

Save As - zapis przekonwertowanego sampla. Format w jakim będzie zapisana próbka ustala się klikając wcześniej na jeden z czterech gadżetów: RAW, IFF, WAV, VOC.

Compress - ustawienie tego gadżetu powoduje, że przed zapisaniem sample zostaną skompresowane. Funkcja ta odnosi się tylko do formatu IFF, pakowanie innych formatów jest nieopłacalne.

Flip - tej funkcji używamy w przypadku konwersji sampli formatu RAW pochodzących z PC lub Macintosha. Format ten różni się trochę od używanego przez Amigę, na przykład brak dźwięku w PC to 0, a na Amidze to 128. Użycie tej opcji powoduje konwersję sampla na odpowiedni format.

Rate - w tym okienku podawana jest ilość odtwarzanych próbek na sekundę. Możemy tutaj podać własną wartość.

Play - kliknięcie na ten gadżet spowoduje odegranie sampla.

Aby przekonwertować daną próbkę przyniesioną na przykład od kolegi, który ma PC, należy (oczywiście po konwersji pliku na format AmigaDos) załadować ją funkcją Open, następnie klikając na Play odsłuchać i w razie potrzeby użyć opcji Flip. Kolejną czynnością jest wybranie jednego z czterech formatów i zapis sampla na dyskietce funkcją Save As. W przypadku plików IFF można jeszcze włączyć opcję Compress. Program jest szczególnie przydatny dla tych amigowców, którzy nie posiadają samplera, ale mają kolegę z PC i Sound Blasterem.

Autorem programu jest Syd L. Bolton.

4. SMPlay V1.0

To ciekawy odtwarzacz sampli. Od innych programów tego typu różni się tym, że może odtwarzać próbki zapisane w czterech formatach: standardowych Amigowskich IFF i RAW oraz pecetowych WAV i VOC. Program uruchamia się tylko z poziomu CLI, z podaniem odpowiednich parametrów:

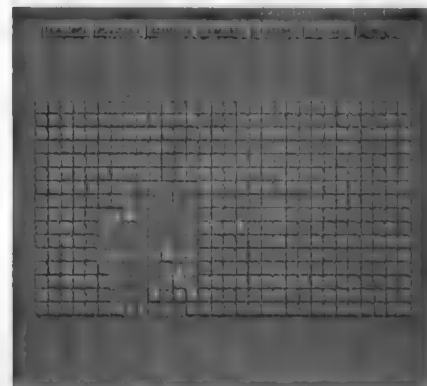
SMPlay nazwa pliku rXXXXX v lX i

Przy czym "rXXXXX" to prędkość odtwarzania sampla w próbkach na sekundę. Ustawienie parametru "v" spowoduje wypisanie informacji o samplu, "lX" to ilość powtórzeń danej próbki (na przykład l2 spowoduje dwukrotne odegranie sampla), parametr "i" ma takie samo działanie jak opcja Flip w programie Sound Machine i może być użyty w przypadku odtwarzania sampli formatu RAW, pochodzących z PC lub Macintosha. Dowolny parametr oprócz oczywiście nazwy pliku można pominąć.

Autorem programu jest Syd L. Bolton.

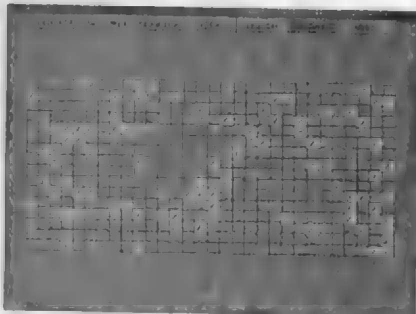
5. Army Miner V1.4

Army Miner czyli saper, to znana logiczna gra planszowa. Zabawa polega na oczyszczeniu pola minowego z niebezpiecznych niespodzianek ukry



wających się pod kwadracikami. No to zaczynajmy!

Klikamy lewym guzikiem myszy na wybrany kwadracik i po chwili odsłania się pewna liczba pól. Jednocześnie na każdym wolnym polu, obok jeszcze nieodsłoniętego pojawia się cyferka oznaczająca ilość min sąsiadujących z tą kratką. Z tych cyferek można wywnioskować położenie min. Jeżeli już takową minę zlokalizowaliśmy, należałoby ją rozbroić. W tym celu na podejrzanym polu klikamy prawym guzikiem myszki, co powoduje postawienie parafki oznaczającej minę. Jeśli zmieniliśmy zdanie, co do położenia ładunku, parafkę zdejmujemy klikając jeszcze raz prawym guzikiem na zaznaczonym polu. Należy uważać, gdyż kliknięcie lewym guzikiem myszki na zaminowane pole powoduje wybuch i śmierć drogiego nam sapera. Jeśli rozminujemy całe pole minowe w krótkim czasie, to komputer pochwali nas mówiąc "Good work son.", a następnie będziemy mogli zapisać swoje dane w poczet najlepszych saperów.



Teraz przedstawię więcej szczegółów. Na ekranie oprócz pola gry znajdują się gadżety:

NEW GAME - rozpoczęcie nowej gry.

PAUSE - zatrzymanie gry.

UNDO - działanie tego gadżetu przeczy zasadzie, że saper myli się tylko raz. Jednak to drobne odstępstwo od reguły może być pomocne w przypadku, gdy po prostu pomyliliśmy się. Użycie UNDO powoduje zwiększenie licznika czasu o 10 sekund.

Oprócz wyżej wymienionych na ekranie znajdują się jeszcze gadżety wyboru stopnia trudności:

BEGINNER - dla początkujących.

ADVANCED - dla zaprawionych w rozminowywaniu.

EXPERT - tylko dla weteranów.

Poszczególne poziomy różnią się wielkością pola oraz ilością min. Jeżeli jednak żaden z wyżej wymienionych poziomów nie będzie nam odpowiadał, możemy używając opcji CUSTOM samemu ustalić długość i szerokość pola minowego oraz ilość znajdujących się na nim niespodzianek.

Poza gadżetami na ekranie, do dyspozycji mamy górne menu. Oto jego opis:

PROJECT - tutaj oprócz funkcji dostępnych z gadżetów mamy dodatkowo informacje o programie (ABOUT) oraz wyjście (QUIT).

OPTIONS - w tym menu możemy utrudnić bądź ułatwić sobie grę:

Use questions marks - włączenie tej opcji powoduje, że klikając dwukrotnie prawym klawiszem myszy na wybranym polu stawiamy tam pytajnik, który jest ignorowany przez program, ale może być nam pomocny.

Safe [illegible] - opcja bezpiecznego startu. Jej aktywność powoduje, że pierwsze kliknięcie na planszę nigdy nie spowoduje wybuchu.

Auto mark - włączenie tej funkcji szalenie upraszcza grę. Po prostu gdy klikniemy na odkryte już pole, komputer sam zaznaczy miny lub odkryje wolne kratki (o ile jest to możliwe).

Safe klick - działanie tej opcji objawia się tym, że jeżeli klikniemy na wybrane pole, a następnie trzymając guzik opuścimy je, to powróci ono do pierwotnego stanu.

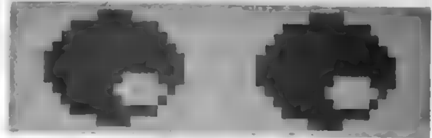
Sound effects - włączenie lub wyłączenie efektów dźwiękowych.

SCORES - tutaj możemy obejrzeć (Show) lub skasować (Clear) wyniki najlepszych saperów.

Grę można uruchamiać z poziomu Workbencha lub CLI, ale w tym drugim przypadku należy powiększyć pamięć dla stosu komendą stack 30000. Uruchamianie z CLI ma tę zaletę, że podając parametr początkowy dla generatora pseudolosowego, grę można uruchamiać kilka razy z tak samo zaminowaną planszą. Szczerze zachęcam do grania w sapera, gdyż niesamowicie rozwija logiczne myślenie i wyobraźnie, a zwycięstwo (bez sztuczek) na najwyższym poziomie jest powodem do dumy. Grajmy więc, pamiętając, że prawdziwy saper myli się tylko raz!

Autorem programu jest Alain Laferriere.

6. Follow Mouse V1.2



Po uruchomieniu programu, na ekranie pojawia się para oczu, które mrugają do nas sympatycznie. Co więcej oczka te wodzą komputerowym wzrokiem za kursorem myszki. Gdy nie dotykamy naszego zwierzątka przez dłuższą chwilę, to (chyba z nudów) zapadają w sen.

O przydatności Follow Mouse decyduje jedynie poczucie humoru użytkownika. Mnie osobiście programik przypadł do gustu. Na dyskietce znajduje się również kod źródłowy w Kick Pascalu.

7. Coolhosen

To moduł muzyczny napisany przez Dagona z grupy Zero Zone.

8. Listingi programów..

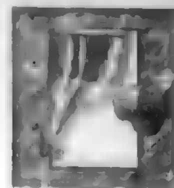
W tej szufladzie znajdziecie listingi do wszystkich kursów języków programowania prowadzonych w Amigowcu.

Autorem większości ikonek zamieszczanych na naszych dyskietkach jest Dariusz Zwierzyński.

Prosimy o uwagi o naszej bibliotece. Piszcie do nas, jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy też na owoce Waszej pracy.

Jak zamawiać i ile to kosztuje ?

Dyskietki PD można zamawiać przesyłając pieniądze na nasze konto, przy pomocy blankietu z Amigowca, podając symbol dysku PD (od #1/92 do 12/92, #1/93, #2/93, #3-4/93, #5-6/93, #7-8/93, #1/94, #2/94, #3/94 oraz "A", "B" i "C") i oczywiście swój czytelny adres. Nasza dyskietka kosztuje 48 tysięcy złotych (wliczone są koszty dyskietki, przesyłki, nalepki, opakowania itp.). W przypadku przesyłki za zaliczeniem pocztowym, zamawiający pokrywa koszty zaliczenia. ❑



LISTA DEMONÓW

Artur Łukasik, Sebastian Klomski

Witamy po raz kolejny! Wszyscy czekamy na wiosenne PARTY w Starachowicach, które odbędzie się w dniach 16-17 kwietnia 1994. Zjedzie się tam cała polska scena, obudzona po zimowym śnie.

ZAGRANICA
GRUPY:

1. **Sanity**
2. **Kefrens**
3. **Virtual Dreams/Fairlight**
4. **Razor 1911**
5. **Complex**
6. **Spaceballs**
7. **Melon Dezign**
8. **Movement**
9. **Lemon**
10. **Silents**

Po PARTY III w Danii rozpadło się kilka doskonałych grup, między innymi międzynarodowa grupa Lemon składająca się z byłych członków grupy Anarchy. Jej członkowie zasilili grupę Razor 1911, która jest w tej chwili największą konkurencją niezmordowanego Sanity. Z niecierpliwością czekamy na ich nowe demo, które z pewnością będzie charakteryzowało się doskonałą grafiką (nowo nabyci graficy z Lemonu) i rewelacyjną muzyką (prawdopodobnie autorstwa jednego z najlepszych muzyków na scenie - Lizard Kinga). Pozostali przyłączyli się do Spaceballs (Dan-ko der i Facet-grafik) i innych grup, ■ najlepszy koder (Hannibal), muzyk (Spa ceman ex. Nuke) i grafik (J.A.D.E.) pracują dla firmy Core Design Ltd. tworzącej wspaniałe gry.

Szok wywołała wiadomość o rozłamie w jednej z najlepszych grup ■■ świecie jaką jest (była) grupa Kefrens. Kilku Kefrenów założyło nową grupę o interesującej nazwie Polka Brothers. Nie mniej jednak lider (najlepszy koder i muzyk) pozostał w Kefrensach.

DEMA:

1. **Hardwired** ***/Cronics ■ The Silents***
2. **Desert Dream*/Kefrens***
3. **Arte*/Sanity***
4. **Full Moon*/Fairlight***
5. **Extension*/Pygmy Projects***
6. **Origin*/Complex***
7. **9 Fingers*/Spaceballs***
8. **World of Commodore*/Sanity***
9. **State o The Art*/Spaceballs***
10. **Guardian Dragon II*/Kefrens***

Wydawać by się mogło, że wszystkie lepsze zagraniczne grupy zapadły na pewien czas w odrętwienie. Są to tylko pozory, gdyż wszyscy szykują niespodzianki na kolejne PARTY.

DEMA PLIKOWE:

1. **Interference*/Sanity***
2. **Mindriot*/Andromeda***
3. **3D Demo II*/Anarchy***
4. **Paradigma*/Complex***
5. **Announce*/Lemon***
6. **Transpiersky*/The Silents***
7. **Clairvoyance*/Absolute***
8. **Terminal Fuckup*/Sanity***
9. **Joyride*/Phenomena***
10. **Blue House II*/Rebels***

INTRA 40 KB:

1. **Tetris*/Melon Dezign***
2. **Virtual Dreams**
3. **Shining**
4. **Lemon**
5. **Arrilfoll** ***/Dual Crew ■ Shining***
6. **Fairlight**
7. **Stellar**
8. **The Silents**
9. **No Pain No Gain*/TRSI***
10. **Spaceballs**

MUSIC DISKI:

1. **Memorial Songs*/Alcatraz***
2. **Chromagic** ***/Dual Crew ■ Shining***
3. **Jesterday*/Sanity***
4. **Mirror*/Andromeda***
5. **Turmoil*/Sanity***
6. **Crystal Symphonies II** ***/Phenomena***
7. **Music Dream II*/Phenomena***
8. **Kuglep0len*/Static Bytes***
9. **Book of Song*/Complex***
10. **Crystal Symphonies I** ***/Phenomena***

KODERZY:

1. **Chaos*/Sanity***
2. **Laxity*/Kefrens***
3. **Mr.Pet*/Sanity***
4. **The Spy*/Crionics***
5. **Tsunami*/Fairlight***
6. **Lone Starr*/Spaceballs***
7. **Microforce*/Sanity***
8. **Dr.Skull**
9. **Performer*/Melon Dezign***
10. **Hannibal*/ex. Lemon***

GRAFICY:

1. **Cougar*/Sanity***
2. **Facet*/Lemon***
3. **Ra*/Sanity***
4. **R.W.O.** ***/Polka Brothers***
5. **Mack*/Melon Dezign***
6. **Peachy*/Masque***
7. **Hof*/Melon Dezign***
8. **Rack*/Absolute***
9. **Walt*/Melon Dezign***
10. **Titan*/Complex***

MUZYCY:

1. **Lizard King/*Razor 1911***
2. **Jester/*Sanity***
3. **Audiomonster/*Melon Dezign***
4. **Moby/*Sanity***
5. **Laxity/*Kefrens***
6. **Nuke/ex. Lemon**
7. **Tip & Mantronix/*Phenomena***
8. **Jesper Kyd/*The Silents***
9. **Clawz/*Complex***
10. **Virgill/*Masque***

POLSKA
GRUPY:

1. **Old Bulls**
2. **Deform**
3. **Applause**
4. **W.F.M.H.**
5. **Alchemy**
6. **Suspect**
7. **Joker**
8. **Damage**
9. **Beta Team**
10. **TRSI PL**

Czołówka polskich grup bez większych zmian. Sytuacja ta zmieni się zapewne po oczekiwanym przez wszystkich PARTY organizowanym przez Illusion, Termos i Beta Team w dniach 16-17 kwietnia 1994 w Starachowicach. Ponadto we wrześniu w Bydgoszczy odbędzie się PARTY organizowane przez miejscowe grupy (między innymi Old Bulls). Organizatorzy przewidują bardzo wysokie nagrody pieniężne i rzeczowe (rzędu dziesiątek milionów złotych, czyli porównywalne z wyróżnieniami na PARTY na zachodzie) oraz wiele niespotykanych nigdy wcześniej atrakcji.

Polska sekcja międzynarodowej grupy Mystic zaczyna się rozrastać i jej skład na dzień dzisiejszy wygląda następująco:

XTD - muzyk
NINJA - swaper
MARS - swaper
CHARON - koder
KATANI - koder
JACKAL - grafik
GUNMAN - grafik

DEMA:

1. **Impulse 2/*TILT/Old Bulls***
2. **Deformations/*Deform***
3. **Ray World/*Deform***
4. **Fugazi/*Old Bulls***
5. **PRO.s.i.a.k/*FCI***
6. **Hallucinations & Dreams /*Union***
7. **Software/*W.F.M.H.***
8. **Marchewki/*Alchemy***
9. **Technological Death/*Mad Elks***
10. **The Return/*Joker***

Na pierwszych dwóch miejscach tradycyjnie dema Impulse 2/TILT/Old Bulls i Deformations/Deform.

Na miejscu trzecim wreszcie pojawiło się Ray World/Deform, czyli pierwsze polskie demo na A1200.

Nowe dema pojawią się dopiero po Starachowickim PARTY, ponieważ wiele dobrych grup już od dawna szykuje swoje produkty. Z całą pewnością grupa Old Bulls uraczy nas kilkoma smakowitymi kąskami.

DEMA PLIKOWE:

1. **Party-zanci/*Joker***
2. **Bara Bara/*Applause***
3. **Impulse-preview/*Old Bulls***
4. **40-k intro/*Alchemy***
5. **X-Mas/*Damage***
6. **Autobiography/*Old Bulls***
7. **Varathron/*Suspect***
8. **Xenium/*Old Bulls***
9. **No Name/*Suspect***
10. **Defene coś tam/*Luzers***

DYSKI MUZYCZNE:

1. **Orgasm/*Old Bulls***
2. **Vengeance/*Beta Team***
3. **Waiting for Another Beer/*Joker***
4. **Hunt for the red kret/*FCI***
5. **Kadi Sound Disk III/*Joker***
6. **Illusion/*Old Bulls***
7. **Kadi Sound Disk II/*Joker***
8. **Music Tracks/*Suspect***
9. **Lost Tracks/*Illusion***
10. **Sigh/*Old Bulls***

KODERZY:

1. **Musashi/*Union***
2. **Hudi/*TILT/Old Bulls***
3. **Bukka/*TILT/Old Bulls***
4. **Vico/*Alchemy***
5. **Robin/*W.F.M.H.***
6. **Falcon/*Applause***
7. **Miklesz/*Damage***
8. **Dalthon/*Joker***
9. **Tom/*TRSI PL***
10. **Kane/*Suspect***

MUZYCY:

1. **XTD/*Union***
2. **Peters/*Joker***
3. **Pic Saint Loup**
4. **Accord/*Gel Dezign/Old Bulls***
5. **Dreamer/*FCI***
6. **Mr.Root/*Union***
7. **Phobos/*Beta Team***
8. **Kadi/*Joker***
9. **Snoopy/*ex. Union***
10. **Passat/*Funzine***

GRAFICY:

1. **TPP/*Applause***
2. **Animal/*TRSI PL***
3. **Sabe/*TILT/Old Bulls***
4. **Seq/*TRSI PL***
5. **Zefir/*Gel Dezign/Old Bulls***
6. **Pluton/*TRSI PL***
7. **Frodo/*Alchemy***
8. **Python/*TRSI PL***
9. **Rys/*Joker***
10. **Berserker/*Investation***

MAGAZYNY DYSKOWE:

1. **Thing/*Applause***
2. **Zasmashka/*Alchemy***
3. **Zig Zag/*Union***
4. **Paper White/*Luzers***
5. **X-mag/*Union***
6. **Poczytaj Mi Mamo**
7. **Fat Agnus/*Investation***
8. **Nie z tej beczki/*Lamers***
9. **Krawężnik/*Status OK***
10. **Imazine/*Saint Group***

Tak jak można było się spodziewać doskonały magazyn Thing śląskiej grupy Applause wkroczył na pierwsze miejsce. Jest to spowodowane lenistwem konkurencji:

- Zasmashka nie wychodzi już od pół roku;

- Zig Zag - ostatni numer ukazał się rok temu i prawdopodobnie już więcej numerów nie ujrzy światła dziennego. Zobaczymy czy jego następca X-mag dorówna temu wspaniałemu magazynowi.

Pozostałe magazyny wyraźnie odbiegają od poziomu jaki prezentuje Thing.

Wszystkie wiadomości pochodzą z pewnych źródeł i są w 100 procentach sprawdzone.

Głosujemy na dowolną ilość pozycji z każdego działu, przyznając im od 1 do 10 punktów.

Głosy, propozycje, swoje prace, uwagi, a także nowe programy demonstracyjne i magazyny dyskowe prosimy przesyłać na adres redakcji. ❑



PODSUMOWANIE POLSKIEJ SCENY DEMO

Artur Łukasik, Sebastian Klomski

Przeszłość lubi podsumownia. Także i my postanowiliśmy obiektywnie podsumować dotychczasową działalność krajowych grup demo. W związku z tym podliczyliśmy wyniki wszystkich PARTY (tabelka na końcu artykułu).

Pierwsze prawdziwe party w historii polskiej sceny amigowej odbyło się 16 listopada 1991 roku w Gdynii. Zorganizowały je grupy JOKER i LUZERS. W ówczesnym czasie należały one do ścisłej czołówki krajowych grup. Na party pojawiło się wiele nowych grup, które zaprezentowały jak na tamte czasy całkiem niezłe produkcje. Po raz pierwszy udało się przeprowadzić competition, czyli konkurs na najlepsze demo, muzykę i grafikę. Zadebiutowało wiele wybitnych postaci sceny np. XTD (najlepszy z dzisiejszych muzyków), Pic Sanit Loup, Mr.Root, Animal (najlepszy obecnie grafik), Seq. Na PARTY zostały zaprezentowane takie dema jak:

The Return/*Joker* - niesamowicie dopracowane pod względem graficznym.

Power of Inspiration/*H.O.T.* - pierwszy w Polsce wektorowy świat.

Defene coś tam/*Luzers* - dobre demo plikowe.

Sigh/*Old Bulls* - dysk muzyczny, udany debiut bydgoskiej grupy.

Magic Code/*Action Direct* - głównie popis kodera.

Demo/*Addonic* - ciekawe i dość pomysłowe efekty.



Mega Demo/*Zack Team* - zdecydowanie najgorsze z wystawionych dem.

Po demo competition dwie grupy: H.O.T. i Zack Team połączyły się w jedną, którą nazwały Deform.

Drugie party odbyło się 28 marca 1992. w Warszawie. Pomimo, iż organizatorzy (Action Direct, G-Force, Katharsis) nie zadbali o wiele ważnych spraw, to można powiedzieć, że PARTY było jednym z najlepszych w historii. Gościły na nim takie osobistości światowej sceny jak Chaos i Mr.Pet z doskonałej, niemieckiej grupy Sanity. Każdy z uczestników imprezy miał możliwość rozmowy (oczywiście jeżeli znał język angielski lub niemiecki) z bezsprzecznie najlepszymi ludźmi sceny na świecie.

Demo competition stało na najwyższym poziomie w historii polskiej sceny. Czołówka przedstawiała się następująco:

Deformations/*Deform* - obrazki wektorowe i tańczące postacie (takie jak w State of the Art/Spaceballs).

Marchewki/*Alchemy* - świetna muzyka oraz transformowane obiekty z 2048 punktów.



Faster then Hell/*W.F.M.H.* - najlepszy w tamtych czasach TV BOX na świecie.

Hexadecimal Perfections/*Action Direct* - całkiem niezłe megademo.

Testament/*Katharsis* - głównie efekty wektorowe.

Xenium/*Old Bulls* - rozbudowane demo plikowe.

Mimo, że brakowało identyfikatorów, competition muzyczne było nieprzemyślane, ginęły dyski i ogólnie organizatorzy nawalili, wielu uczestników PARTY wspomina je bardzo mile.

Najmniejsze [redacted] wielkich spotkań sceny odbyło się 27 czerwca 1992 roku w Żywcu. Party zostało zorganizowane przez polską sekcję międzynarodowej grupy Grace. Organizacja była zadowalająca, chociaż bez technicznych fajerwerków. Nie przyjechali ludzie z tzw. elity, demonstrując w ten sposób swoją domniemaną (urojoną) wyższość nad pozostałą częścią sceny.

Wystawiono następujące dema:

Immortal Visions/*Suspect* - dobra muzyka.

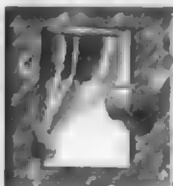
Damage/*Old Bulls* - wektorowa wycieczka po Żywcu.

Hallucinations [redacted] Dreams/*Katharsis* - dynamiczny teledysk techno.

Fly with [redacted] through stars/*Pepsi Drinkers Inc.* - do oglądania przez okulary trójwymiarowe.

Poza konkursem zostało wystawione dopracowane demo Party-zanci grupy Joker. Niestety z powodu niedyspozycji przedstawiciela Jokerów (niejakiego Petersa) nie znalazło się na competitions, choć z pewnością miało szansę na wysoką pozycję.

Demo Hallucinations [redacted] Dreams /*Katharsis* mimo, iż było tworzone przez osoby związane [redacted] tą grupą (Tom - kodowanie, Mad Max - muzyka, Python - grafika), nie było firmowane jej nazwą.



Drugie warszawskie PARTY odbyło się [redacted] listopada 1992 roku. Organizacja mimo drobnych niedociągnięć była bardzo dobra. Można powiedzieć, że żadne wcześniejsze [redacted] późniejsze PARTY nie było [redacted] udane. Swoje prace zaprezentowało wiele nowych dotychczas nieznanych grup (np. Flying Cows Inc., Mad Elks, Defensywa, Beermacht, Barbie Fuckers).

Zostały wystawione następujące dema:

Software/*W.F.M.H.* - niesamowita muzyka.

Fugazi/*Old Bulls* - bardzo dopracowane graficznie.

Alternative Reality/*Suspect* - niezła muzyka w połączeniu [redacted] nienajgorszymi pomysłami.

Hunt for Red Kret/*Flying Cows Inc.* - połączenie dema z dyskiem muzycznym.

Weekend Dance/*Deform* - samplowana muzyka i grafika wektorowa w trybie HAM.

Status Quo/*Mad Elks* - pierwsze demo poznańskiej grupy.

Hit the Sky/*Investation* - ładny raytracing.

Beton/*Defensywa* - debiut pilskiej formacji.

Beer/*Beermacht* - grafika wektorowa transformująca się w rytm muzyki.

Delirium/*Deform* - oświetlane wektorowe boby.

Barbie Fuckers - dowcipne demko napisane w Amosie.





Poza standardowymi konkursami (czyli najlepsze demo, muzyka, grafika) odbyły się także inne jak np. competition na najlepszą aranżację muzyki ze starego polskiego serialu telewizyjnego pt. "Stawka większa niż życie" [redacted] na najlepszego kierowcę w grze Lotus firmy Gremlin.

W dniach 1-3 maja 1993 roku miał miejsce Mountain Congress zorganizowany przez grupę Applause. Wystawiane dema były [redacted] większości jeśli nie debiutem grupy, to jej pierwszą poważną produkcją. W związku [redacted] tym poziom nie był najwyższy.

Oto wyniki demo compo:

Technological Death/*Mad Elks* - efekty zgrane z muzyką.

Hallucinations [redacted] Dreams/*Union* - długo zapowiadane trackmo.

Hydra/*Investation* - demo [redacted] efektami w stylu produkcji z przed kilku lat.





Varathorn/*Suspect* - pierwsze pliko[redacted] demo gdańskiego teamu.

Neurasthenia/*Hiron* - dość ciekawa muzyka.

[redacted] Demo/*Credo* - bardzo zabawna karykatura prawdziwego dema.

Kill the Dutchman/*Proxis* - całkiem udany powrót na scenę.

Prototype/*Beta Team* - pierwsze demo grupy zajmującej się dotychczas dyskami graficznymi.

Psycho Medium [redacted]/*Turnips* - druga część psychodelicznego dema.

Impulse preview/*Old Bulls* - zapowiedź dema.

Dampax/*Damage* - dużo raytracowanej grafiki.

Amsey/*Brainiax* - debiut nowej gdańskiej grupy.

First Comming/*Blasphemy* - niezbyt udane demo.

Proffanation/*Omega [redacted] Black Bit* - bardzo nudne trackmo.

Poza tym ukazało się kilka pozakonkursowych produkcji:

Raytraced Drams/*Old Bulls*
Betag III/*Beta Team*
Vengeance/*Beta Team*
Orgasm/*Old Bulls*

oraz następujące magazyny dyskowe:

Krawężnik I
Zig Zag VII
Fat Agnus [redacted]

W dniach 23-24 października 1993 roku odbyło się w Poznaniu amigowe party zorganizowane przez miejscowe grupy (Mad Elks, Flying Cows Inc., Dioxide, Lesiu Independent). Po raz pierwszy nagrody były wręczane nie tylko za pierwsze miejsca, ale także za drugie i trzecie.

Wyniki demo competition były następujące:

Impulse/TILT/*Old Bulls* - niesamowite efekty, niesamowicie zgrane z niesamowitą muzyką.



Pro-siak/*Flying Cows Inc.* - demo z dobrymi efektami i świetną muzyką.

Uquala/*Damage* - demo dopracowane pod względem graficznym.

Overvision/*Blaze* - kilkuczęściowe debiutanckie trackmo .

Ponadto zostało zaprezentowane bezkonkurencyjne (z powodu braku innych dem [illegible] A1200) trzydyskowe demo grup Deform i Investation. Ray World (bo tak brzmiała nazwa tego demka) na big screenie wyglądał oszałamiająco. Jednak po obejrzeniu na zwykłym monitorze, nie robi wielkiego wrażenia. ❑

Punktacja [illegible] grup za poszczególne miejsca zajęte w demo compo.

Lp. grupa	Gdynia 16.11.1991.	Warszawa 28.03.1992.	Żywiec 27.06.1992.	Warszawa 6.11.1992.	Żywiec 1.05.1993.	Poznań 23.10.1993.	Podsumowanie
01.OLD BULLS	4	6	2	2	10	1	42
02.DEFORM	2	1	-	5	-	*	25
03.SUSPECT	-	-	1	3	4	-	25
04.W.F.M.H.	-	3	-	1	-	-	18
05.FLYING COWS INC.	-	-	-	4	-	2	16
06.MAD ELKS	-	-	-	6	1	-	15
07.KTS	-	4	3	-	-	-	15
08.ACTION DIRECT	5	5	-	-	-	-	12
09.INVESTATION	-	-	-	7	3	-	12
10.JOKER	1	-	-	-	-	-	9
12.DAMAGE	-	-	-	-	11	3	9
13.UNION	-	-	-	-	2	4	7
17.HIRON	-	-	-	-	5	-	6
18.ADDONIC	6	-	-	-	-	-	5
19.CREDO	-	-	-	-	6	-	5
20.AKWILON	-	7	-	-	-	-	4
21.PROXIS	-	-	-	-	7	-	4
22.ZACK TEAM	7	-	-	-	-	-	3
24.DEFENSYWA	-	-	-	8	-	-	3
25.BEERMACHT	-	-	-	9	-	-	2
26.TURNIPS	-	-	-	-	9	-	2
27.BARBIE FUCKERS	-	-	-	10	-	-	1
28.BLASPHEMY	-	-	-	-	13	-	1
29.BRAINIAX	-	-	-	-	12	-	1
30.ILLUSION	-	-	-	-	14	-	1

Przyjęliśmy następujący sposób punktowania: za pierwsze miejsce - 10 punktów, za drugie - 9, za trzecie -8 itd. Za dziesiąte i każde kolejne po jednym punkcie.

** - grupie Deform została przyznana nagroda [illegible] najlepsze demo w kategorii A1200. Jako że było to jedyne demo na A1200, nie można tego traktować jako zwycięstwo.*



BOBY i Sprite'y

Tomasz [illegible]

Ten odcinek poświęcony będzie zastosowaniu w programach bobów i sprite`ów, czyli dwóch grup ruchomych obiektów graficznych oferowanych przez komputery Amiga. Spotykamy je najczęściej w grach, jako wiele małych obiektów (postaci) poruszających się po ekranie na tle innej grafiki. Jak łatwo zauważyć, ich użycie pozwala na uzyskanie naprawdę ciekawych efektów, które można w prosty sposób wykorzystać programując w Amosie

Na początku zajmę się sprite'ami. Są to objekty umieszczane na ekranie przez Coppera i poddane pewnym ograniczeniom wynikającym z konstrukcji Amigi. Standardowy sprite sprzętowy ma szerokość 16 i wysokość (maksymalnie) 255 punktów. Możemy utworzyć osiem takich sprite'ów w trzech kolorach i cztery w piętnastu kolorach (kolor nr 0 jest traktowany jako przezroczysty). Nie jest to wiele, ale przy umiejętnym zastosowaniu i wykorzystaniu dodatkowych możliwości oferowanych przez Amosa uzyskuje się całkiem nie złe wyniki. Uwaga!!! nieco inaczej wyglądają te ograniczenia w Amigach wyposażonych w kości AGA, lecz jest to temat na inny artykuł. Tym razem skupię się na tym, co można uzyskać na wszystkich modelach Amigi. To tyle tytułem wstępu, dalsze informacje przedstawię przy opisywaniu poszczególnych komend.

Sprite n,x,y,o

n - numer sprite`a
x,y - współrzędne
o - numer obrazu z banku sprite'ów

Komenda ta wyświetla sprite'a o numerze n i obrazie o w miejscu o współrzędnych x,y. Przed wykorzystaniem tej instrukcji należy pamiętać o umieszczeniu w odpowiednim banku pamięci obrazów sprite'ów. Są one przechowywane w stałym, zgrywanym razem z programem banku nr 1. Można wypełnić go obrazami, stosując edytor sprite'ów znajdujący się na dysku z Amosem, wczytać gotowy bank komendą Load "tytuł",1 lub też użyć instrukcji Get Sprite, o której mowa w dalszej części artykułu. Wszystkie obrazy przechowywane w takim banku są ponumerowane i wybierając stosowny numer (parametr o) decydujemy, który obraz ma stać się spritem o numerze określonym przez parametr n. Parametry tego obrazu (wymiary, liczba kolorów) decydują o późniejszych parametrach sprite'a, o czym nie należy zapominać przy umieszczaniu obrazów w banku pamięci. Można umieścić tam obraz, z którego nie da się stworzyć sprite'a i zamiast pożądanego efektu zobaczymy komunikat o błędzie. Na pocieszenie mogę podać, że Amos w pewnym stopniu usuwa ograniczenia, o których wspomniałem na początku artykułu. Każdy sprite może mieć szerokość od 16 do 128 punktów i wysokość od 1 do 255 punktów, 15 kolorów (kolor 0 - przezroczysty), a ich liczba teoretycznie może wynosić 63. W tej pięknej wizji jest tylko jedno "ale". Sprite'y o większej liczbie kolorów, szerokości, czy numerach powyżej 7 powstają przez łączenie i dzielenie na segmenty podstawowych ośmiu sprite`ów sprzętowych. Łatwo więc wyobrazić sobie sytuację w której na ekranie znajdują się obiekty, do powstania których wykorzystane zostały wszystkie sprite'y sprzętowe. W takiej sytuacji próba wprowadzenia kolejnego obiektu na ekran skończy się niepowodzeniem. Wynika z tego niezbicie, że tworząc np. szerokiego, piętnastokolorowego sprite'a można zużyć wszystkie sprite'y sprzętowe i pozbawić się możliwości umieszczenia równocześnie na ekranie innego sprite'a. Współrzędne położenia sprite'a określa się względem tzw. "gorącego punktu" znajdującego się zazwyczaj w jego lewym, górnym rogu (choć można to zmienić) i podawane są jako współrzędne sprzętowe, czyli niezależne od trybu ekranu (rozdzielczość). Aby móc łatwiej umieszczać sprite'y na ekranie, należy przeprowadzać konwersję tych współrzędnych na współrzędne ekranowe za pomocą specjalnych funkcji, przedstawionych w dalszej części artykułu.

Get Sprite Palette

Instrukcja ta ładuje paletę ekranu kolorami z palety wykorzystywanej przez obrazy sprite'ów. Jest to w zasadzie konieczne przed wprowadzeniam ich na ekran, gdyż używana przez nie paleta jest zazwyczaj inna niż paleta ekranu, z której pobierane są kolory do wyświetlania wszystkich obiektów na nim się znajdujących. Jeśli więc nie zastosuje się tej instrukcji, można doprowadzić do "przekłamań barw" co jest efektem bardzo nieprzyjemnym.

Set Sprite Buffer n

n - wysokość sprite'ów

Jest to instrukcja pozwalająca zaoszczędzić nieco, tak potrzebnej nieraz pamięci. Ustala ona maksymalną wysokość sprite'ów na wartość podaną przez parametr n. Jeśli więc żaden z naszych sprite'ów nie jest wyższy niż np. 30 punktów. Możemy bez żadnej szkody zdefiniować maksymalną wysokość wszystkich sprite'ów na 30 i zwolnić przez to część bufora przeznaczonego do ich przechowywania. Jeśli jednak któryś obraz będzie wyższy niż podana wartość, zostanie odpowiednio przycięty.



X Screen(n,x)

n - numer ekranu
x - współrzędna sprzętowa x

Funkcja ta przyjmuje wartość współrzędnej ekranowej (ekranu o numerze n) punktu określonego współrzędną sprzętową x.

Y Screen(n,y)

n - numer ekranu
y - współrzędna sprzętowa y

Funkcja analogiczna do poprzedniej, lecz odnosząca się do współrzędnych pionowych (y).

X Hard(n,x)

n - numer ekranu
x - współrzędna ekranowa x

Funkcja dokładnie odwrotna do X Screen. Zamienia współrzędną ekranową na sprzętową.

Y Hard(n,x)

n - numer ekranu
x - współrzędna ekranowa y

Działanie jak wyżej dla współrzędnej y.

X Sprite(n)

n - numer sprite'a

Funkcja przyjmująca wartość współrzędnej sprzętowej x sprite'a o numerze n.

Y Sprite(n)

n - numer sprite'a

Funkcja podobna do poprzedniej, lecz odnosząca się do współrzędnej y. Wraz z X Sprite pozwala na sprawdzanie czy np. poruszający się sprite zajął daną pozycję lub opuścił ekran.

Get Sprite e,n,x1,y1 to x2,y2

e - numer ekranu
n - numer obrazu w banku sprite'ów
x1,y1 x2,y2 - współrzędne prostokąta

Instrukcja ta przenosi z ekranu o podanym numerze prostokąt określony współrzędnymi x1,y1 i x2,y2 do banku sprite'ów, jako obraz o numerze n.

Sprite Off n

n - numer sprite'a

Usuwa z ekranu sprite'a o podanym numerze, a jeśli parametr n zostanie pominięty, usuwa wszystkie sprite'y.

Teraz chciałbym zaprezentować kilka instrukcji pozwalających na "poruszenie" naszych sprte'ów. Będzie to jednak tylko kilka prostych ruchów, gdyż programowaniu animacji i kontroli nad obiektami ruchomymi planuję poświęcić jeden z następnych odcinków. Zajmę się w nim szczegółowo AMAL'em, czyli stanowiącym integralną część Amosa specjalnym językiem przeznaczonym do oprogramowania obiektów ruchomych i animacji.

Move X n,a$

n - numer obiektu np. sprite'a
a$ - zmienna tekstowa, zawierająca definicję ruchu

Komenda ta pozwala na zdefiniowanie poziomego ruchu sprite'a o numerze n. Definicja zawarta w zmiennej a$ musi mieć postać trzech liczb oddzielonych przecinkami i umieszczonych w nawiasie okrągłym np. (1,5,30). Pierwsza liczba określa ilość okresów o długości 1/50s jakie mają upłynąć między dwoma kolejnymi krokami ruchu. Druga liczba określa liczbę punktów o jaką ma być przesuwany obiekt w każdym kroku ruchu. Jeśli jest dodatnia, obiekt będzie przemieszczał się w prawo, a jeśli ujemna w lewo. Trzecia liczba określa liczbę powtórzeń zdefiniowanego przez dwie pierwsze wartości kroku ruchu. Ustawienie liczby powtórzeń na 0 powoduje powtarzanie ruchu w nieskończoność. Na końcu listy może zostać użyte jedno z dwóch jednoliterowych oznaczeń. "L" powodujące ciągłe powtarzanie całej sekwencji ruchów i "E" pozwalające zatrzymać obiekt w punkcie o podanej współrzędnej x. Definicja prostego ruchu może wyglądać np. tak:

Move X 1,"(1,5,30)E100"

Sprite nr 1 będzie przesuwany co 1/50s o 5 punktów w prawo, przy czym przesunięcie zostanie wykonane 30 razy. Jeśli w czasie ruchu obiekt znajdzie się w pozycji o współrzędnej x=100 zostanie zatrzymany, bez względu na liczbę wykonanych kroków.

Move Y n,a$

n - numer obiektu
a$ - definicja ruchu

Jest to instrukcja analogiczna do poprzedniej, lecz przesuwająca obiekt w pionie. Dodatnie wartości kroku (liczby punktów przesunięcia) powodują ruch w dół.

Move On / Off

Włącza lub wyłącza ruch zdefiniowany dwoma poprzednimi instrukcjami. Wbrew pozorom, te dwie instrukcje pozwalają na zdefiniowanie całkiem ciekawych efektów, ponieważ w każdej z nich umieścić można dość długą listę ruchów, a łączenie ze sobą przemieszczania w pionie i poziomie, powoduje przesuwanie "na ukos". Dodatkową zaletą tych instrukcji jest wykonywanie ich z wykorzystaniem systemu przerwań, czyli przemieszczanie obiektów nie zatrzymuje programu w Basicu, ani nie blokuje pracy edytora. Najlepiej będzie obejrzeć to na przykładzie "Ruchome sprite'y". Po jego wpisaniu, trzeba oczywiście umieścić w banku pamięci obrazy sprite'ów. Najprościej jest wczytać z dysku (komendą Load "tytuł",1) gotowy bank zawierający potrzebne obrazy np. Sprites.abk z dysku Amosa. Jak można się przekonać, sprite'y będą poruszać się bez względu na dalsze operacje wykonywane przez nasz program lub użytkownika (radzę wpisać kilka poleceń w trybie bezpośrednim).

```
Rem Ruchome sprite'y

Screen Open 0,320,256,32,Lowres
Get Sprite Palette
Curs Off
Sprite 1,140,60,1
Sprite 2,140,70,1
Sprite 3,140,80,1
Move X 1,"(1,2,135)(1,-2,135)L"
Move Y 2,"(1,2,100)(1,-2,100)L"
Move X 3,"(1,2,100)(1,-2,100)L"
Move Y 3,"(1,2,100)(1,-2,100)L"
Move On
Direct
End
```

Teraz już czas najwyższy zająć się bobami. Są to obiekty umieszczane na ekranie przez blitter i ogólnie rzecz biorąc, można posługiwać się nimi tak, jak sprite'ami. Nie ma jednak ograniczeń charakterystycznych dla sprite'ów. Wymiary bobów nie są ograniczone, a liczba dostępnych kolorów jest taka jak w przypadku ekranów. Liczba bobów jest wstępnie ustalona na 63, lecz można ją zwiększyć przy użyciu programu zmieniającego konfigurację Amosa. Trzeba jednak pamiętać, że bank zawierający obrazy bobów zajmuje sporo pamięci. Pomimo nie



zaprzeczalnej przewagi nad sprite'mi, za ujemne cechy bobów można uznać właśnie większe zapotrzebowanie na pamięć i nieco wolniejsze odtwarzanie.

Bob n,x,y,o

n - numer boba
x,y - współrzędne
o - numer obrazu z banku bobów

Komenda ta rysuje boba o numerze n w miejscu o współrzędnych ekranowych x,y. Pamiętać trzeba, aby przy umieszczaniu na ekranie bobów stosować współrzędne ekranowe, a nie sprzętowe, tak jak to było w przypadku analogicznej instrukcji dotyczącej sprite'ów. Parametr o jest numerem obrazu z banku bobów. Jest to bank nr 1, czyli ten sam, który jest używany do przechowywania obrazów sprite'ów. Obrazy te można wykorzystywać zarówno do tworzenia bobów, jak i sprite'ów, jeśli oczywiście w przypadku tych drugich pozwalają na to parametry obrazu (wymiary, liczba kolorów). Przed umieszczeniem bobów na ekranie dobrze jest, tak jak przy sprite'ach, zastosować instrukcję Get Bob Palette, która i tym razem doskonale spełni swoje zadanie.

Double Buffer

Komenda ta usuwa bardzo nieprzyjemny efekt migotania, powstający przy poruszaniu się po ekranie dużej ilości bobów. Tworzy się on w wyniku wyświetlania na ekranie bobów w trakcie rysowania. Są one rysowane bowiem w tym samym czasie co ekran i przy wielu obiektach, komputer po prostu "nie zdąży" narysować wszystkich bobów przed odświeżeniem ekranu. Usunięcie tego zjawiska polega na utworzeniu "podwójnego bufora ekranu", czyli drugiego "logicznego" ekranu, na którym rysowane są boby. Dopiero po utworzeniu wszystkich obiektów, jego zawartość jest zamieniana z zawartością ekranu "fizycznego" (widocznego), ■ rysowanie bobów rozpoczyna się ponownie na ekranie "logicznym". Proces ten jest wykonywany całkowicie automatycznie i nie zakłóca żadnych innych operacji. Po wydaniu komendy Double Buffer można więc zapomnieć o problemach związanych z odświeżaniem bobów. Jedynym ujemnym skutkiem stosowania "podwójnego bufora" jest podwojenie zapotrzebowania na pamięć, potrzebną do przechowywania danych ekranu.

Bob Clear

Usuwa wszystkie boby z ekranu i odtwarza tło pod nimi zawarte.

Bob Draw

Odtwarza na ekranie boby usunięte poprzednią instrukcją.

X Bob(n)
Y Bob(n)

n - numer boba

Funkcje analogiczne do X Sprite i Y Sprite. Przyjmują wartość odpowiedniej (x lub y) współrzędnej ekranowej boba o podanym numerze.

Limit Bob n,x1,y1 To x2,y2

n - numer boba
x1,y1 x2,y2 - współrzędne prostokąta

Instrukcja ta ogranicza obszar, w którym widoczny jest bob o numerze n do prostokąta o wymiarach określonych przez współrzędne x1,y1 i x2y2. Jeśli numer boba zostanie pominięty, instrukcja będzie dotyczyć wszystkich bobów wprowadzonych na ekran. Widoczność bobów na całym ekranie przywraca komenda Limit Bob bez parametrów.

Get Bob e,n,x1,y1 To x2,y2

e - numer ekranu
n - numer obrazu w banku
x1,y1 x2,y2 - współrzędne

Komenda działająca identycznie jak Get Sprite.

Put Bob n

n - numer boba

Komenda ta kopiuje na ekran (na stałe) boba o numerze n, w miejscu, w którym dany obiekt znajduje się w chwili wykonywania tej instrukcji.

Paste Bob x,y,n

n - numer boba
x,y - współrzędne

Kopiuje na ekran (na stałe) obraz o numerze n z banku bobów w miejsce o współrzędnych x,y.

Tym razem to wszystkie podstawowe informacje na temat bobów. Jak zwykle na koniec przedstawię praktyczne zastosowania opisanych instrukcji. Będą to trzy związane ze sobą programiki.

"Konwerter rysunków" pozwala na utworzenie banku pamięci zawierającego 64 boby powstałe przez "pocięcie" wczytanego rysunku w formacie IFF. Po uruchomieniu programu w standardowym oknie wyboru pliku trzeba wybrać nazwę obrazka. Musi on mieć rozdzielczość 320*256 i 32 kolory. Następnie przez kolejne "wycinanie" prostokątów od prawego górnego do lewego dolnego rogu, utworzony zostaje odpowiedni bank pamięci. Można go nagrać na dysk po wpisaniu nazwy w kolejnym oknie wyboru, które ukaże się po wykonaniu "konwersji".

```
Rem Konwerter rysunków

F$=Fsel$("*.*","","Wybierz rysunek !!")
If F$="" Then Edit
Load Iff F$,0
C=1
For B=0 To 224 Step 32
For A=0 To 280 Step 40
Get Bob 0,C,A,B To A+40,B+32
C=C+1
Next A
Next B
F$=Fsel$("*.*","","Wpisz nazwę banku !!")
Save F$,1
Erase 1
Cls
End
```

Dwa kolejne programiki prezentują możliwości wykorzystania tak przygotowanego banku pamięci. "Efekt spirali" powoduje pojawienie się obrazu na ekranie, przez ukazywanie się kolejnych prostokącików w sposób sprawiający wrażenie "zwijania się spirali" (prędkość tego procesu zmienić można przez odpowiednie dobranie wartości zmiennej "W"). Jest to efekt dobrze znany użytkownikom programu SCALA. Po wpisaniu programu trzeba wczytać przygotowany wcześniej bank pamięci. Następnie najlepiej przejść do trybu bezpośredniego edytora (klawisz ESC) i wpisać polecenie Load "nazwa_banku",1. Procedura ta mimo swej prostoty daje naprawdę ciekawy efekt i po włączeniu do innego programu może go bardzo uatrakcyjnić (np. spowodować efektowne pojawienie się planszy tytułowej).

"Układanka" jest to prosta GRA. Polega na ułożeniu z małych fragmentów całego obrazu. Po uruchomieniu programu



ukazuje się okno wyboru pliku pozwalające na wczytanie banku pamięci przygotowanego wcześniej "Konwerterem rysunków". Następnie przez chwilę pokazywany jest cały obraz wczytanego rysunku z naniesioną kratą wskazującą na linie podziału. Po zniknięciu obrazu mamy do dyspozycji czarny ekran i "przyczepiony" do wskaźnika myszki fragment obrazka. Należy go przesunąć w odpowiednie miejsce i pozostawić wciskając lewy klawisz myszki. Po przyciśnięciu klawisza spacji fragment zmieni się na inny, losowo wybrany z grupy 64 stanowiących cały obraz. W ten sposób należy ułożyć kompletny obrazek. W przypadku trudności z wyborem odpowiedniego miejsca można wywołać na ekran "gotowy" obraz wciskając prawy klawisz myszki. Gierka ta jest bardzo prosta, ale myślę, że zainteresuje zwolenników łamigłówek.

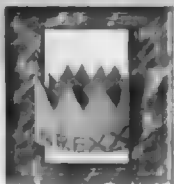
```
Rem Efekt spirali

Screen Open 0,320,256,32,Lowres
Get Sprite Palette
Flash Off
Hide
W=1
A=280
B=224
C=240
D=192
E=32
F=0
G=0
S=1
For P=0 To 3
For X=G To A Step 40
Paste Bob X,F,S
S=S+1
Wait W
Next X
S=S+7
For Y=E To B Step 32
Paste Bob A,Y,S
S=S+8
Wait W
Next Y
S=S-9
For X=C To F Step -40
Paste Bob X,B,S
S=S-1
Wait W
Next X
S=S-7
For Y=D To E Step -32
Paste Bob G,Y,S
S=S-8
Wait W
Next Y
A=A-40
B=B-32
C=C-40
D=D-32
E=E+32
F=F+32
G=G+40
S=S+9
Next P
End
```

```
Rem Układanka

Screen Open 1,320,256,32,Lowres
Load Fsel$("*.*","","Wybierz plik !!"),1
Get Sprite Palette
Hide
C=1
For B=0 To 224 Step 32
For A=0 To 280 Step 40
Paste Bob A,B,C
Box A,B To A+40,B+32
C=C+1
Next A
Next B
Wait 20
Screen Open 0,320,256,32,Lowres
Get Sprite Palette
Flash Off
Double Buffer
Randomize Timer
Hide
A=1
Do
If Mouse Key=2
While Mouse Key=2
Screen To Front 1
Wait 5
Wend
Screen To Front 0
End If
X1=X Screen(X Mouse)
Y1=Y Screen(Y Mouse)
For K=0 To 280 Step 40
If K<X1 and X1<K+40 Then X=K
Next K
For K=0 To 224 Step 32
If K<Y1 and Y1<K+40 Then Y=K
Next K
If Inkey$=" "
A=Rnd(64)
If A=0
A=1
End If
Wait 5
Clear Key
End If
Bob 1,X,Y,A
If Mouse Key=1
Put Bob 1
Wait Vbl
End If
Loop
End
```

Na zakończenie życzę wszystkim powodzenia w programowaniu i dobrej zabawy przy "rozpracowywaniu" naszej pierwszej gierki. Zachęcam także do jej udoskonalenia. Można dołączyć muzyczkę, planszę tytułową (spirala), licznik czasu, ilości wywołań obrazu pomocniczego, itd. Do zrobienia jest jeszcze sporo, więc można wykazać się sporą inwencją. Artykuł ten z pewnością przybliżył czytelnikom Amigowca podstawowe zasady tworzenia bobów i sprite'ów, lecz jak łatwo zauważyć jest to temat bardzo obszerny i jak już wspomniałem jeszcze do niego wrócę. Następnym razem zajmę się tworzeniem typowego menu rozwijanego z górnej belki. Będzie to bardzo przydatne w naszych, coraz bardziej rozbudowanych programach. ❑



AREXX cz.4

Izabela Skibińska

W dzisiejszym odcinku będziemy mieli okazję docenić walory Arexxa w zakresie sterowania programami z zewnątrz na przykładzie znanego programu do obróbki graficznej obrazów - Art Department Professional.

Nie przez przypadek wybór padł na ten właśnie program. Brak automatyzacji uczyniłby jego użytkowanie czynnością wręcz przerażającą w przypadkach takich, jak jednakowa obróbka klatek animacji, gdy mozolnie musielibyśmy przeprowadzać te same operacje na każdym obrazku. Jak wiadomo z poprzednich odcinków, aby ustalić komunikację pomiędzy programem ADPro a interpreterem Arexxa, korzystamy z rozkazu ADDRESS z argumentem "ADPro", będącym nazwą arexxowego portu programu ADPro.

Zanim przejdziemy do omawiania przykładów, parę słów uzupełnienia o samej komendzie ADDRESS. Dotychczas poznaliśmy jej jednoargumentową wersję (ADDRESS portname) oraz dwuargumentową (ADDRESS portname command) nie zmieniającą aktualnego "host" portu. Istnieje trzecia forma tego rozkazu, bezargumentowa (tzn. ADDRESS) powodująca zamianę portu aktualnego z poprzednim. Arexx zarządza bowiem jednocześnie dwoma portami. Gdy nie ustalimy żadnego "host" portu, aktualnym jest port interpretera Arexxa, a poprzednim port dla komend AmigaDos. W przypadku zdefiniowania nowego "host" portu, aktualny staje się poprzednim, natomiast poprzedni ginie. Dwuargumentowa forma rozkazu ADDRESS naturalnie nie wywołuje żadnej zamiany. Możliwość poznania aktualnego portu daje nam funkcja address().

Wracając do Adpro - program ten ma możliwość uruchomienia 50 skryptów arexxowych znajdujących się w katalogu REXX:, przy czym obowiązują określone zasady nazewnictwa plików. Przyciskając klawisze funkcyjne (F1-F10) spowodujemy wywołanie programów o nazwach od F1.adpro do F10.adpro. Analogicznie, przyciskając Fx z klawiszem SHIFT, uruchamiamy skrypty SFx.adpro, Fx z ALT - LFx.adpro, Fx z klawiszem Amiga - AFx.adpro, Fx z Ctrl - CFx.adpro. W przypadku, gdy interpreter nie znajdzie odpowiedniego skryptu wysyłany jest stosowny komunikat. Rezultaty poleceń wykonywanych przez ADpro są uzyskiwane tak jak w innych programach, przez komendę OPTIONS RESULTS, z tą jednak różnicą, że wynik działania komendy umieszczany jest w zmiennej arexxowej ADPRO_RESULT. W przypadku wystąpienia błędu (tzn. gdy RC ~=0) znajdować się tam będzie jego opis.

Na początek skrypt sprawdzający obecność ADPro w systemie. W przypadku nie znalezienia nastąpi próba uruchomienia programu. Zauważmy, że konieczne jest dokonanie przypisania ADPRO: do katalogu, gdzie znajduje się program. Obecność owego przypisania testowana jest funkcją statef z rexsupport.library, która w przypadku znalezienia obiektu określonego argumentem (w naszym przypadku jest to katalog określony przez przypisanie ADPRO:) zwraca łańcuch zawierający takie informacje jak typ obiektu, długość, itp., natomiast, gdy dany obiekt nie istnieje, otrzymujemy łańcuch pusty.

W skrypcie Locateadpro.rexx na uwagę zasługuje użycie funkcji time do mierzenia przepływu czasu. Wywołana z parametrem 'R' zeruje zegar, a następnie czas, który upłynął. Od tego momentu można pobrać funkcję time z parametrem 'E'.

Niniejszy skrypt możemy wykorzystać w innych celach, przeznaczonych już wyłącznie do użytkowania ADPro. Nie potrzeba jednak kopiować go każdorazowo do określonego skryptu. Zgodnie z tym, o czym mówiliśmy w poprzednim odcinku, do interpretera Arexxa możemy przesłać zarówno rozkaz jak i nazwę skryptu (Patrz Przykład p1.rexx)

```
/*Locateadpro.rexx - uruchomienie ADPro */

  options results
  maxloadingtime = 60
  Flag = 0
        if (connectlib('rexxsupport.library',-30) == 0) then
      return 0

  if (statef( 'ADPRO:' ) = "") then
              return 0

  time( 'R' )
  do while (time( 'E' ) < maxloadingtime) & (pos( 'ADPro',show
  ( 'Ports' ) ) = 0)
      if (Flag = 0) then do
        address command 'Run < NIL: > NIL: ADPRO:ADPro '
```



```
            Flag = 1
                end
        address command 'Wait 1'
            end
if (pos( 'ADPro', show( 'Ports' ) ) = 0) then
    return 0
        else
return 1
        exit

connectlib: procedure
parse arg libname,offset

if ~show(Libraries,"rexxsupport.library ") then do
  parse value "libs:"libname with fullname

    if exists(fullname) then do
      addlib(libname,0,offset,0)
    return 1
    end
    else
    return 0
  end
```

Uwaga! Tekst wytłuszczony należy wpisywać w jednej linii!

Najistotniejszymi rozkazami Arexxa dla programu ADPro są IFORMAT - określający format ładowanego obrazka, SFORMAT - format w jakim ma zostać zapamiętana grafika, OPERATOR - działanie na obrazku oraz EXECUTE - zamiana 24-bitowych danych, na których dokonywana była obróbka graficzna na dane do wyświetlenia z parametrami określonymi przez Screen Controls (liczba kolorów do wyświetlenia, rozdzielczość).

```
/* p1.rexx */

if Locateadpro() then do

  address "ADPro"
        options results
  adpro_to_front
  lformat "IFF"
  sformat "IFF"

  getfile '"Select file"'
  filename=adpro_result
  if ~exists(filename) then
            exit
  else do
            load filename
    getstring '"Select operator"' "COLOR_TO_GRAY"
    op=adpro_result
  if op="COLOR_TO_GRAY" then do
      okay2 "Additional arguments ?"
      if rc~=0  then do

    getnumber '"Enter weight for red"' 1000 0 10000
              rw = adpro_result
    getnumber '"Enter weight for green"' 1000 0 10000
              gw = adpro_result
    getnumber '"Enter weight for blue"' 1000 0 10000
              bw = adpro_result

              operator op rw gw bw
            end
        else
          operator op
            end
    else
          operator op
    execute
    filename=insert(filename,"converted")
    save filename image
        okay1 "Operation completed"
    end
  end
```

W skrypcie p1.rexx zastosowałam typowo amigowy format ładowania i zapamiętywania grafiki (IFF). Równie dobrze parametrami mogłyby być nazwy innych modułów odpowiedzialnych [redacted] wczytywanie i zgrywanie obrazków (PCX, TIFF, itp).

Choć w przypadku rozkazów LFORMAT i SFORMAT obowiązuje dokładnie jeden argument, sytuacja przedstawia się znacznie gorzej przy komendach LOAD i SAVE, których argumenty zależą od formatu grafiki. Na przykład przy wielu formatach jako częstej opcji przy rozkazie LOAD używa się opcji NOPAD, która powoduje to, iż obrazek dla którego nie ma 24 bitowych danych (lub [redacted] bitowych dla szarości) nie zostanie przekonwertowany przez ADPro na 24 bitowe dane. Przyspiesza to w znaczny sposób ładowanie takich grafik.

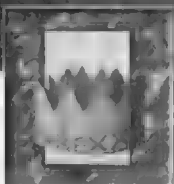
Przykładem obowiązkowego użycia wielu argumentów dla rozkazu LOAD jest format BACKLINE (umożliwia tworzenie różnego rodzaju tła). Lista opcji przedstawia się następująco: TYPE W H POS <S> <M> <E> DIR, gdzie TYPE przyjmuje wartości "COLOR" lub "GRAY", W i H są szerokością i wysokością tworzonego obszaru, POS jest pozycją środkowego koloru relatywnie do początkowego i końcowego koloru, <S> <M> i <E> określają wartości kolorów początkowego, środkowego i końcowego (w przypadku szarości jest to intensywność o wartości od 0 do 255), DIR natomiast jest kierunkiem nachylenia przyjmującym jedną z ośmiu wartości ("NW", "N", "NE", "E", "SE", "S", "SW" i "W"). Króciutki przykład:

```
/*F1.adpro */

ADDRESS "ADPro"

S="50 100 150"
M="200 40 40"
E="100 50 200"
GRAY="128 255 128"
 lformat "BACKLINE"
 load "X" "COLOR" 320 256 50 S M E "W"  /*dla koloru*/
execute
 load "X" "GRAY" 320 256 50 GRAY  "E"  /* dla szarości */
execute
```

Wracając do przykładu p1.rexx, bardzo istotny jest parametr rozkazu SAVE, który musi być określony przez jeden z następujących napisów: RAW, IMAGE lub SCREEN. RAW spowoduje zapamiętanie "surowych" danych 24 lub [redacted] bitowych, IMAGE - danych potraktowanych funkcją EXE



CUTE czyli do wyświetlenia, SAVE - zapamiętana zostanie ta część obrazka, która zmieści się w aktualnym screenie.

Arexx dla programu ADPro zawiera sporo elementów ułatwiających kontakt z użytkownikiem. Należą do nich różnego rodzaju requestery. Najprostszy z nich, uzyskiwany rozkazem OKAY1, pozwala na wyprowadzenie komunikatu i jest w postaci OKAY1 "string", gdzie string jest dowolnym ciągiem znakowym. Rozkaz OKAY2 umożliwia udzielenie odpowiedzi twierdzącej lub przeczącej przez użytkownika. Zmienna RC będzie zerem w przypadku wyboru "Cancel", natomiast przy kliknięciu na "OK" przyjmie wartość niezerową.

Pobranie łańcucha znaków od użytkownika uzyskujemy komendą GETSTRING. Pełna forma rozkazu wygląda tak: GETSTRING TITLE DEFAULT, gdzie TITLE jest tytułem requstera, a DEFAULT ciągiem znakowym już w nim zawartym. Podobnie przedstawia się definicja rozkazu GETNUMBER z tym, że dwa dodatkowe argumenty oznaczają tu minimalną i maksymalną akceptowaną wartość.

Do wyboru plików służy rozkaz GETFILE o pełnej składni: GETFILE TITLE DEFDIR DEFFILE, której już chyba nie trzeba opisywać.

Bardzo ważną rzeczą w requesterach wywoływanych rozkazami GETSTRING, GETFILE i GETNUMBER jest konieczność zawarcia ciągu znakowego zawierającego spacje w podwójnych cudzysłowach (Patrz Przykład p1.rexx).

Atrakcyjną formę requestera plikowego umożliwiającą wielokrotny wybór, a tym samym ułatwiającą mechanizację zadań, możemy uzyskać rozkazem GETFILES, gdzie wyboru wielu plików z jedego katologu dokonujemy przy wciśniętym klawiszu SHIFT. Przykładowy skrypt znajduje się w pierszym odcinku kursu o Arexxie. Istotną sprawą jest zawarcie listy plików, oddzielonych spacjami w zmiennej ADPRO_RESULT. Otrzymaną listę możemy analizować funkcjami przeznaczonymi do operacji na słowach (Word() i Words()).

Kolejny skrypt służy do obróbki plików wchodzących w skład jednej animacji.

```
  /* F4.adpro */
  /* odwraca animacje "do góry nogami " */

address 'ADPro'
options results

NL = '0A'X   /* nowa linia */

lformat "anim"
sformat "anim"

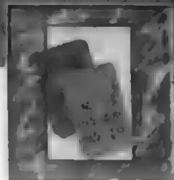
pstatus "UNLOCKED"

adpro_to_front
getfile '"Enter anim to flip"' "ADPRO:"
source_anim = adpro_result
if (rc ~= 0) then do
    okay1 '"No source anim"'
    exit
end
else do
  getfile '"Enter anim file to create"' 'ADPRO:'
  if (rc~=0) then
      dest_anim=insert(source_anim,"dest")
    dest_anim = adpro_result
    okay2 '"Flip animation? "'
    if (rc=0)
    then  do
       address command 'Copy 'source_anim dest_anim
       okay1 '"Thank you for playing"'
       exit
     end
  /* Flipping */
    load source_anim "FRAME" 1
    pstatus "LOCKED'
    load source_anim "COUNT"
    NumFrames = adpro_result
    getnumber '"Enter start frame "' 1 1 10
    StartFrame = adpro_result
    getnumber '"Enter end frame "' 12 1 12
    EndFrame = adpro_result
    if (EndFrame < StartFrame +2) then
      exit 10
    currframe = startframe

    do while (currframe < EndFrame)
      load source_anim "FRAME" currframe
      if (rc~=0) then do
 okay1 "Error during load" || NL ||,
"of frame '|| currframe'."
 exit 10
        end
        operator "VERTICAL_FLIP"
        execute
        save dest_anim "APPEND" "IMAGE"
        currframe = currframe+1
    end
   save dest_anim "QUIT' 'IMAGE"
end
```

Idea skryptu jest bardzo prosta. Po określeniu animacji źródłowej i ilości klatek do zamiany, w pętli dokonujemy operacji na poszczególnych klatkach i dołączamy je do nowo tworzonej już odwróconej animacji. Pewną niejasność mogą stanowić argumenty przy komendach LOAD i SAVE. Zgodnie z tym co napisałam wcześniej, mają one bezpośredni związek z formatem. Tak więc LOAD z opcją COUNT, zwróci nam liczbę klatek animacji, natomiast podając parametry postaci FRAME n spowodujemy wgranie n-tej klatki. Z kolei rozkaz SAVE z opcją APPEND dołącza kolejne klatki do nowo tworzonej animacji, parametr QUIT "zamyka" plik z animacją. Rozkaz PSTATUS z opcją UNLOCKED spowoduje ustawienie palety takiej, jaka jest we wgrywanej grafice. Po wgraniu pierwszej klatki "zamrażamy" paletę, a tym samym powodujemy, że utworzona animacja będzie posiadała jeden zestaw kolorów.

W następnym odcinku znów będziemy zajmować się ADPro i nie tylko. ❑



CrossDOS

Jarosław Ch[illegible]owski

CROSSDOSa znamy wszyscy - jest integralną częścią systemu operacyjnego Amigi. Nie trzeba nikogo przekonywać że jest ułatwiającym życie narzędziem. To przy jego pomocy można bez problemów przenosić dane pomiędzy systemami AmigaDOS, a MS-DOS. Pod nazwą CROSSDOS kryje się cały pakiet, zawierający m.in. programowy emulator IBM XT - CROSSPC v2.23.

Pakiet obejmuje 1 dyskietkę z programami CROSSDOS i CROSSPC oraz z programikami pomocniczymi: instalacyjnymi, diagnostycznymi i wspomagającymi działanie CROSSDOSa. Ciekawostką jest dołączenie programu "anty-instalującego", czyli usuwającego pliki związane z CROSSDOSem. Wreszcie ktoś pomyślał o tym, by nie robić śmietnika z naszych twardych dysków.

Uruchomienie CROSSDOSa jest proste - wystarczy kliknąć na ikonę 'PC0', by stacja df0: potrafiła odczytywać dane MS-DOSa. Jeśli chcemy posiadać stację PC0: również po resecie, należy przekopiować plik 'PC0' do katalogu devs:dosdrivers. Ta procedura jest identyczna z uruchamianiem CROSSDOSa dostarczanego z oprogramowaniem systemowym. Sam CROSSDOS (jako program, nie jako pakiet) nie różni się bowiem wiele od tego, którego mamy w systemie.

Nowa wersja CROSSDOSa pozwala na odczytywanie dysków zapisanych z ilością 10 sektorów na ścieżkę (kłania się ATARI). Pełna lista systemów rozpoznawalnych przez CROSSDOSa wygląda teraz następująco:

- MS-DOS: 360 KB, 720 KB, 1.44 MB
- ATARI: 720 KB
- Floptical 20 MB

Ciekawostką jest rozpoznawanie stacji dysków optycznych. Dyski takie mogą być widziane jako MS-DOS 720 KB, MS-DOS 1.44 MB lub też można je sformatować we własnym systemie (lub AmigaDOSie) - wtedy mamy do dyspozycji 20 MB.

Formatowanie dysków pod MS-DOSem jest proste: wystarczy kliknąć na ikonę formatowanego dysku w stacji PC0: (należy uważać, bowiem dyskietka w stacji DF0:/PC0: posiada dwie ikony!) i wybrać opcję FORMAT DISK... z menu ICONS na Workbenchu. Czyli wszystko po staremu.

Uruchomienie emulacji PC również nie wymaga większej filozofii, wystarczy - hmm - "dwukliknąć" na ikonę CROSSPC, by uruchomić emulację PC-XT. Teraz należy jedynie włożyć dyskietkę z systemem do stacji i można rozpoczynać pracę.

CROSSDOS i CROSSPC pozwalają na korzystanie z twardego dysku. Istnieją tu dwie metody pracy: korzystanie z napędów sformatowanych pod MS-DOSem oraz emulacja twardego dysku PC na partycji Amigi (w pierwszym przypadku autorzy zalecają używanie przy konfiguracji napędów wcześniej sformatowanych w MS-DOSie). Po skonfigurowaniu napędu od strony PC należy skorzystać z poleceń FDISK oraz FORMAT, co jest zresztą standardową procedurą.

W porównaniu z wcześniejszymi wersjami CROSSPC wprowadzono m.in. następujące zmiany:

- zwiększono szybkość emulacji portu szeregowego. Bez problemów można teraz wykorzystywać na CROSSPC takie programy komunikacyjne jak Procomm, Telix, Compuserve Information Manager, Prodigy, Laplink.
- zwiększono szybkość obsługi twardego dysku przez MS-DOS o około 10%.
- mamy do dyspozycji emulację drivera myszki, co pozwala wykorzystywać większość aplikacji PC.
- program konfigurujący twardy dysk pod MS-DOS został ulepszony, można teraz przy jego pomocy wykorzystywać dyski nie formatowane jako RigidDiskBlock.

Plik MOUSE.COM pozwala na wykorzystanie standardowej myszki Amigi jako myszy peceta. Można teraz posłużyć się po prostu kombinacją klawiszy 'prawa AMIGA - HELP', by od razu uruchomić driver. Wyłączenie drivera jest realizowane kombinacją 'prawa AMIGA - SHIFT - HELP'. Należy jedynie pamiętać, by uaktywnić driver myszki przed uruchomieniem aplikacji.

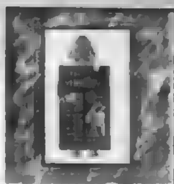
Niemiłą rzeczą jest konieczność uruchamiania programu DiskChg po każdej zmianie dyskietki w stacji 5.25" PC. Jest to spowodowane brakiem sygnału zmiany dyskietki, tak więc sygnał należy wysłać własnoręcznie.

CROSSDOS i CROSSPC pozwalają na korzystanie z dysków zapisanych w formacie MS-DOS. Problemy mogą się jednak pojawić, gdy oba programy będą żądać dostępu w tym samym czasie. Aby zabezpieczyć się przed takimi konfliktami, zastosowano standardowy sposób blokowania dostępu do dysku. Otóż CROSSPC korzystając z urządzeń PC0: lub PCC: blokuje dostęp do urządzenia od strony Amigi. Istnieje co prawda możliwość jednoczesnego wykorzystywania urządzenia od strony Amigi (CROSSDOS) i PC (CROSSPC), jednak twórcy programu nie zalecają tego początkującym użytkownikom ze względu na możliwość uszkodzenia dysków.

Uruchomienie CROSSPC powoduje otwarcie ekranu przeznaczonego dla IBMa. Emulowaną kartą graficzną jest zabytek architektoniczny w postaci CGA. Szybkość pracy emulatorów programowych nigdy nie była rewelacyjna. Uruchomiony na A4000/040 emulator do przeszukiwania pliku wielkości ok. 30 KB (Norton Commander, tryb View, opcja Search, szukanie nieistniejącego ciągu znaków) potrzebował około 5-6 sekund.

Podczas pracy z CROSSPC czeka nas niemiła niespodzianka: użycie kursora w wysokiej rozdzielczości powoduje, że w momencie przejścia na inny ekran Amigi kursor ten zostaje zniekształcony, na szczęście w sposób nie utrudniający zbytnio życia. Zniekształcenie kursora występuje również w czasie pracy z emulatorem (po uruchomieniu drivera). Pewną radą jest używanie kursora jako sprite'a w niskiej rozdzielczości, lecz komu by się chciało machać takim wielkim "plackiem"...

Posumowując, mamy kolejny programowy emulator Wielkiego Niebieskiego na małą kolorową (żadna nowość) oraz (też żadna nowość) możliwość przenoszenia danych między różnymi systemami. Pakiet CROSSDOS nie oferuje więc nam jakościowo niczego nowego. Po prostu jest. ❑



DROGA Z GAZETAMI

Tomasz Kuleszczyński

Nie przejmujcie się tym, że po pierwszym odcinku naszej Drogi z Gazetami nie wszystko stało się dla Was jasne. To co trzeba powtórzymy inaczej - tak żebyście mogli wreszcie stworzyć Waszą ulubioną gazetkę szkolną o psie Pluto i Gumisiach. (Ostrzegam jednak, że wiedza z poprzedniego AMIGOWCA na pewno się przyda.) No to do dzieła!

Na czym stanęliśmy?

Wiemy jak uruchomić program i jakie opcje dobrać, by przy najmniejszym możliwym mruganiu uzyskać największą możliwą rozdzielczość. Wiemy też jak przygotować sobie polskie czcionki. Ci, którzy nie mają dostępu do spolszczonych czcionek Compugraphic mogą wykorzystać oczywiście istniejące czcionki przygotowane przez innych użytkowników lub samemu spróbować stworzyć namiastkę czcionek przy pomocy programiku Fed [illegible] dysku Extras starszych Amig. Pragnę zauważyć, że program ten jest niedoskonały i nie umożliwia edycji "zbyt" dużych czcionek.

Wiemy również jak "uruchomić" nową stronę z odpowiednimi marginesami. To jednak nie wszystko, aby było ładnie...

Chwila [illegible] wydrukach.

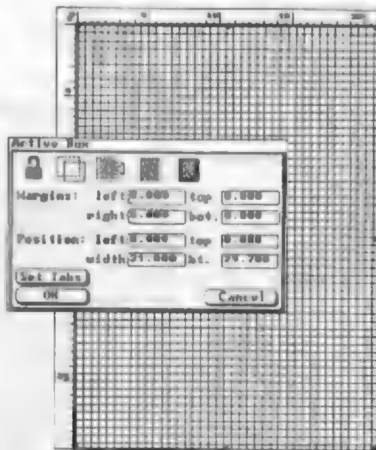
Wydruki na drukarce igłowej sprawią Wam na początku sporo kłopotu. Nie wspominam już o tym, że musicie mieć prawidłowo dobrany sterownik (driver) tak, aby możliwe było drukowanie grafiki. To jest warunek sine qua non - jak mawiają starożytni Rzymianie (czyli podstawowy). Jak to sprawdzić? Wystarczy wydrukować coś z programu Deluxe Paint. Jeśli się udało, to driver jest dobrze dobrany. Teraz wystarczy taki sam sterownik skopiować do odpowiedniej szuflady na dysku (DEVS:printers) i odpowiednio ustawić preferencje. Zajmowaliśmy się tym w AMIGOWCU 7-8/93 [illegible] stronie 19.

Gdy już mamy odpowiednio skonfigurowane preferencje, spróbujmy wykonać jakiś wydruk z naszego programu. Dlaczego musimy to zrobić zanim przystąpimy do tworzenia strony? Dlatego, żebyśmy potem nie ulegli rozczarowaniu, gdy "utną się" nam marginesy naszej przepięknie złożonej gazetki. Program jest na tyle prymitywny, że rządzi się własnymi prawami odnośnie druku. To nic, że margines zostanie ustawiony tak i tak. To akurat nie ma prawie zupełnie znaczenia.

Na samym początku radzę ustawić górny margines na 0 (Top=0). Teraz będziemy mogli powalczyć z marginesami bocznymi i dolnymi. Najłatwiej chyba będzie przygotować sobie stronę wzorcową i dokonać jej wydruku. To co się utnie, utnie się także na innej stronie, [illegible] tak będziemy chociaż uprzedzeni.

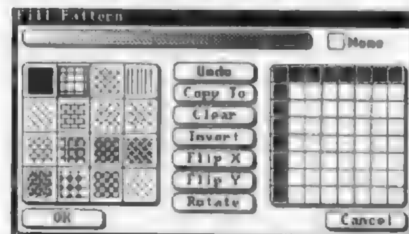
Marginesy boczne

Strona wzorcowa powinna mieć ramkę, którą moglibyśmy sobie zmierzyć linijką, po tym jak nam ją drukarka uciacha. Proponuję coś takiego:



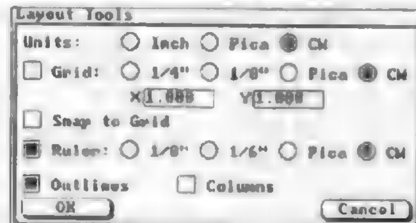
Strona do kalibrowania wydruku.

Jak to zrobić? Jest to niezwykle proste. Przy tworzeniu strony musimy wszystkie marginesy zredukować do zera (left, top, right, bottom) i nacisnąć OK. Oczywiście tworzymy stronę A4. Pojawi nam się strona z odpowiednim pudełkiem rozciągniętym od brzegu do brzegu. Warto by jednak to pudełko czymś wypełnić. W tym celu klikamy na nie 2 razy i ukaże nam się requester "Active Box" (tak, jak to widać na rysunku). W tym requesterze musimy zmienić ostatni gadżet mający formę pustego prostokątu [illegible] gadżet wypełniony (raz kliknąć) - tak, jak to ma miejsce na rysunku. Teraz musimy wybrać odpowiedni wzorek, który da się zmierzyć linijką.



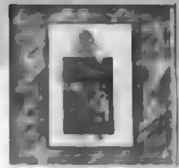
Wzorek może być dowolny, ale [illegible] niszczyć taśmę.

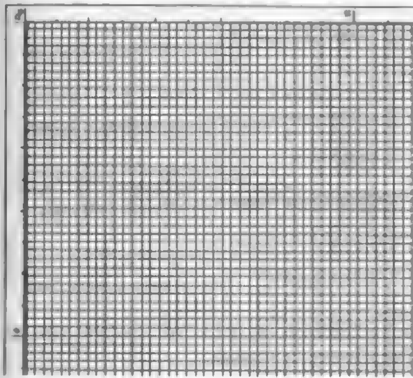
Proponuję wybrać wzorek drugi. Jest to najzwyklejsza w świecie kratka, ale dzięki niej będziemy mogli pomierzyć, ile się nam jej ucięło, a ile zostało. Pomiarów dokonujemy linijką, a następnie wyniki przenosimy na stronę w PageSetterze. Pamiętajmy jednak, żeby w opcji "Layout Tools" ustawić centymerty (CM), a nie cale, bo nic z naszych pomiarów nie wyjdzie.



Ustawiamy CM jako jednostkę miary.

Teraz możemy dokonać próbnego wydruku i już będziemy wiedzieli, [illegible] się nam obcięło. Mierzymy, ile brzegu zostało po stronie lewej i ile po stronie prawej i to będą nasze marginesy. Jednak ustalenie marginesów nie zapobiega wychodzeniu poza nie w trakcie tworzenia strony. Dlatego też [illegible] radzę wykorzystywać podsta





Tak to wygląda [redacted] teorii - pierwsze centymetry.

wowego pudełka, gdyż może się przesunąć w trakcie gorączkowych czynności tworzenia gazety. Trzeba zablokować je zamkiem (kłódeczka w requesterze Active Box) i zostawić na przyszłość. Niestety - przy każdej nowo tworzonej stronie będziemy musieli wpisywać odpowiednie marginesy. Warto zwrócić uwagę na fakt, że można spotkać drukarki drukujące prawie od końca do końca strony, tak więc wielkość marginesów może być diametralnie różna. Marginesy boczne ustawia się jednak najłatwiej.

Margines górny

Jak zapewne zauważyliście u góry mamy pewną stratę. Nie dość tego, że kartka nie może być zadrukowana w całości (w większości drukarek - nawet laserówki zostawiają sobie ok. 0.5 cm na spad), to jeszcze sam program obcina nam powierzchnię druku. Dlatego też górny margines powinien być bezwzględnie ustawiony na zero. Jeśli chodzi o drukarki igłowe, można tam ustawić kartkę tak, żeby wydruk zaczynał się jak najbliżej górnej krawędzi. Oczywiście wymaga to ręcznego cofnięcia kartki. Nie byłoby z tym żadnych problemów, gdyby nie to, że operacja musi być powtarzana - [redacted] każdym razem cofnięcie musi być takie samo. Jak [redacted] zrobić - pozostawiam Waszej inwencji. W każdej drukarce są inne punkty, do których możemy cofać...

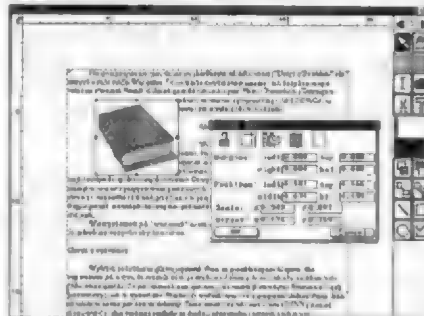
Margines dolny

Ustawienie tego marginesu jest najtrudniejsze. Wielką sztuką jest doprowadzić do tego, żeby wydruk następował nisko jak najniżej. PageSetter po prostu bezczelnie sobie z nas kpi i ucina zbyt dużo. Zostało dowiedzione, że nawet (a właściwie przede wszystkim) przy wydruku na laserówkach zostaje zbyt duży margines. W drukarkach igłowych wystarczy w odpowiednim momencie podłożyć drugą kartkę, tak aby czujnik papieru nie wykazał jego braku. Reszta powinna iść gładko (o ile wydruk nie przejdzie na drugą kartkę lub pierwsza nie zacznie się ślizgać). Natomiast w drukarkach laserowych zaczynają się schody. Okazuje się, że trzeba tu sięgnąć do preferencji i ustawić jak najdłuższą stronę jeśli chodzi o linijki (Paper Lenght (lines) - przy ustawieniu Custom). Jeśli to nie wyjdzie, można zmieniać inne parametry. W sumie jest możliwe uzyskanie dolnego marginesu rzędu niewiele ponad 1 cm. Wymaga to jednak wielu eksperymentów.

W innych programach możecie natknąć się na podobne trudności. Tak sprawuje się niestety ProPage, a wady tej udaje się jedynie uniknąć przy zastosowaniu PostScriptu. Są też na Amigę programy, które ustawianie marginesów realizują bezbłędnie. Należy do nich PageStream, a z programów prostszych, będących pomiędzy edytorem [redacted] DTP - FinalWriter.

Oddziaływania pudełek

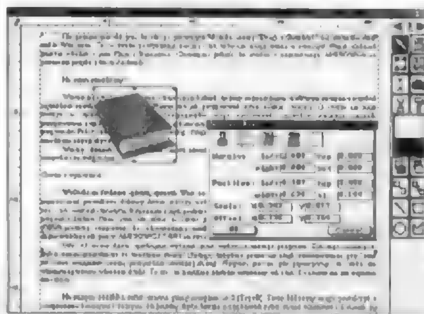
Gdy już mamy gotową stronę, warto by było czymś ją zapełnić. W poprzedniej części "Drogi..." dowiedzieliśmy się jak można importować różnego rodzaju grafiki, czy teksty. Ustawia się pudełko w odpowiednim miejscu i szafa gra. Oczywiście, jak o tym wspomnieliśmy, wszystkie pudełka oddziaływują na siebie wzajemnie. Pudełko z grafiką może zmusić tekst w pudełku pod nim, żeby go opływał.

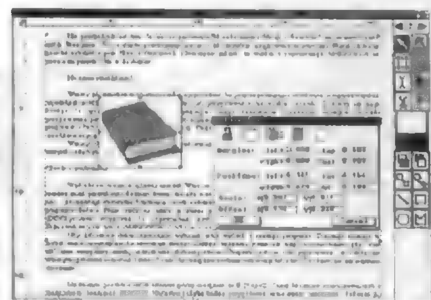


Tekst opływa pudełko..

Nie zawsze uda nam się uzyskać pożądany efekt. Inne programy mają [redacted]żliwość opływania w różny sposób - PageSetter tylko w taki. Zauważyliście tu pewnie jeszcze jedną istotną wadę: zawartość pudełka bezpośrednio sąsiaduje [redacted] tekstem. Oczywiście istnieje możliwość stworzenia marginesiku, ale jest to dosyć mozolne. Trzeba w tym celu wstawić puste pudełko, które ma opływać tekst i nad nim umieścić pudełko właściwe. Oczywiście powstają ogromne problemy przy zmianie wielkości grafiki, czy też przy przemieszczaniu pudełek "tył - przód".

Poza tym grafiki mogą przykrywać tekst tak, aby było widać jego część lub zasłaniać go całkowicie. Ustawienie zależy od stanu drugiej ikony w requesterze Active Box. Oczywiście aktywnym pudełkiem mu-

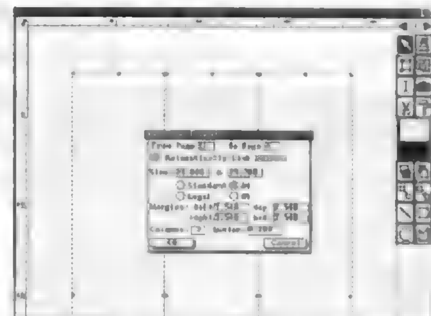




si być odpowiednie pudełko z grafiką, a nie z tekstem.

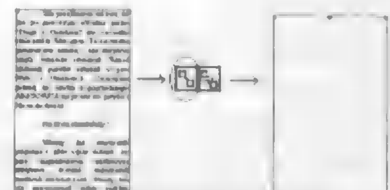
Kolumny

Oprócz pisania [redacted] jednej kolumnie PageSetter umożliwia nam pisanie w kilku kolumnach. Aby stworzyć kolumny musimy zaznaczyć ich ilość przy tworzeniu strony (Columns). W tym samym requesterze zaznaczamy również odstępy pomiędzy kolumnami (Gutter).



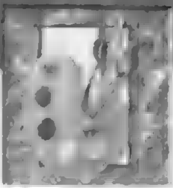
Tak tworzymy kolumny - w tle "gotowa" strona.

Opcja "Automatically Link Columns" spowoduje, że tekst będzie automatycznie przenoszony z jednej kolumny (po jej przepełnieniu) do następnej. Łączenie kolumn pomiędzy stronami odbywa się przy pomocy odpowiedniego gadżetu. Klikamy [redacted] ostatnią kolumnę, następnie na gadżet i na pierwszą kolunę nowej strony lub tę kolumnę, którą chcemy dołączyć do zestawu. Jest to szczególnie istotne przy zmianach wielkości tekstu, gdy właściwie nie wiemy gdzie się skończy.



Gadżety łączenia i [redacted] kolumn.

Na tym etapie możemy zakończyć wprowadzanie do tworzenia gazety. Oczywiście na całą gazetę składają się dobre teksty i grafiki, ale skład nadaje jej ładny wygląd. Z miłą chęcią dowiedziałbym się o waszych wątpliwościach i problemach z dziedziny DTP. Wszelkie pytania kierujcie na adres redakcji. Chętnie zobaczylibyśmy też Wasze dzieła - szkolne gazetki, zaproszenia i inne druczki tworzone na Amidze.



IMAGINE cz.3

Marek Stoc

To już nasze trzecie spotkanie z Imagine i wspaniałym światem grafiki trójwymiarowej. W tym odcinku jeszcze trochę teorii na temat nakładania i stosowania tekstur oraz pędzli. Już za miesiąc startujemy z naszą pierwszą animacją. Wszystkie przykłady robione będą na Imagine 2.0. Wprawdzie coraz bardziej przyzwyczajam się do Imagine 2.9, chociaż ma on niestety za duże wymagania sprzętowe jak na niektóre kieszenie.

TEKSTURY

Nie ma znaczenia jaki jesteś dobry w tworzeniu obiektów, jeśli nie mają one wystarczającej ilości szczegółów, mam na myśli zwłaszcza wygląd powierzchni. Tekstury pozwalają na szybkie stworzenie dla obiektu skomplikowanej powierzchni bez wielkiego nakładu pracy. Rama obrazu wyglądająca jak drewno jest o wiele ciekawsza niż po prostu brązowa. Tekstury są algorytmami, których Imagine używa w celu zdecydowania, jak zamalować dany punkt powierzchni obiektu. Tekstury mogą także wyznaczać przezroczystość, stopień odbicia i kąt w danym punkcie. Mogą one nakładać się dla bardziej skomplikowanych efektów.

Każda tekstura wymaga swoich parametrów. Przeważnie będziesz musiał wprowadzić kolor, przezroczystość i odbicie. Parametry te wprowadza się jako liczby. Ja często używam potencjometrów w requesterze atrybutów, a następnie wpisuję liczby, które muszę. (Pamiętaj o skasowaniu koloru domyślnego!)

Zauważ, że dla większości tekstur, kiedy mówię "kolor", mam na myśli kolor powierzchni, stopień odbicia i przezroczystość. Dla większości tekstur można ustawić wszystkie trzy wartości! Niektóre tekstury jak disturb, będą wpływały na odbicie światła podobnie jak mapa wyskości (altitude brush map). Użycie tekstur wymaga podania dodatkowych parametrów. Może to być wielkość pól szachownicy, gęstość słojów w drewnie lub wielkość cegieł.

Innym parametrem jest oś tekstury, która może być edytowana ręcznie. Oś tekstury jest bardzo ważna. Dla większości tekstur będziesz potrzebował pozycji "bazowej" i orientacji tekstury. Na przykład przy nakładaniu liniowej tekstury program musi wiedzieć, gdzie zacząć i w jakim kierunku iść. Musisz umieścić oś tekstury w odpowiednim miejscu i określić jej kierunek zgodnie z kierunkiem tekstury. Tekstura drewniana jest w najprostszym przypadku grupą koncentrycznych kolorowych walców. Gdzie powinien być środek, w którym kierunku powinien być zwrócony walec? O tym mówi oś tekstury. Niektóre tekstury nie wymagają osi, jak na przykład Camo.

UWAGA! Jeśli oś twojej tekstury znajduje się na ścianie, możesz otrzymać dziwne efekty, ponieważ dla tekstury typu szachownica, granice pól będą wypadały na powierzchni obiektu. Algorytm będzie miał problemy z wybraniem koloru. Efekt ten występuje szczególnie przy nakładaniu na płaszczyznę. Rozwiązanie problemu będzie przesunięcie nieco osi tekstury.

Drewno wygląda najlepiej, jeśli oś jest prawie, ale nie całkiem równoległa do największego wymiaru obiektu. Daje to najbardziej realistyczne efekty.

Wiele tekstur, lecz nie wszystkie, oddziaływuje tylko na niektóre części obiektu. Tekstura kamuflażu jest najlepszym przykładem. Ustawiasz domyślny kolor obiektu z requestera atrybutów. Tekstura Camo następnie nakłada wzorek przez który miejscami prześwituje domyślny kolor obiektu. Podobnie dzieje się przy większości tekstur. Drewno dodaje tylko słoje, a reszta obiektu ma kolor domyślny. Tekstura liniowa stopniowo zmienia kolor z domyślnego na inny. Szachownica zmienia tylko kolor w miejscu jednych pól, podczas, gdy pola koloru przeciwnego mają atrybuty domyślne obiektu.

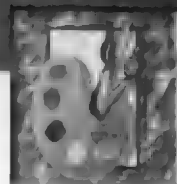
Dlaczego jest to ważne? Dzięki tej własności możesz nakładać do czterech tekstur jednocześnie. Są one nakładane w kolejności od pierwszej do czwartej. Co można w ten sposób zrobić? Możesz stworzyć deskę, nałożyć jako pierwszą teksturę drewno, następnie jako drugą camo. Plamki pokryją drewno, ale tam, gdzie ich nie ma będą nadal widoczne słoje.

Najbardziej użytecznymi teksturami są najprawdopodobniej drewno i tekstura liniowa. Istnieje mnóstwo ciekawych rzeczy, które można uzyskać dzięki nadużywaniu tekstur. Oto jedna z nich:

1. Stwórz obiekt. Dobrze wygląda długi napis. Zamaluj go i nałóż teksturę w dowolny sposób.

2. Dodaj teksturę liniową, ustaw długość przejścia Z na około 20% długości obiektu. Ustaw oś tekstury przy





jednym końcu, obróconą w stronę środka napisu. Zmień kolor tekstury ■ czarny, bez odbić i filtr 255 255 255. Tak, zupełnie przezroczysty. Upewnij się, że tekstura liniowa jest ostatnia, jeśli nakładałeś wcześniej inne tekstury.

3. Włącz renderowanie. Obiekt powinien być niewidoczny, ponieważ tekstura jest zupełnie przezroczysta i pokrywa cały napis. Popraw oś, jeśli wskazywała w złą stronę.

4. Skopiuj obiekt. Przesuń oś tekstury na drugą stronę, zorientowaną w tę samą stronę. Zapisz to z inną nazwą pliku. Sprawdź renderowanie. Powinno to wyglądać jak normalny obiekt, bez tej dziwnej tekstury. Powinno też nie być przezroczyste.

5. W edytorze akcji, przekształć (morph) pierwszy obiekt na drugi. Jedyną zmianą jest oś tekstury, więc Imagine będzie interpolował jej położenie z jednego końca napisu do drugiego, co każdą klatkę. Animacja powinna mieć co najmniej 10 klatek, lepiej 20. Możesz renderować w scanline.

Po napisie będzie "przelatywał" cień. Jest to bardzo impresyjny efekt, ■ zarazem prosty do zrobienia... góra 10 minut.

PĘDZLE ZWANE BRUSHAMI

Tekstury nie są jedynym sposobem na dodanie szczegółów do obiektu. Bardziej bezpośrednia, mniej elegancka, choć bardziej uniwersalna metoda polega ■ używaniu pędzli (brush maps). Brushmapy są standardowymi rysunkami Amigowymi (czasem zwanymi IFFami), które można umieścić na wybranym obiekcie. Możesz nałożyć rysunek swojej twarzy na płaską powierzchnię, dołożyć ramy (z użyciem tekstury drewnianej!) i otrzymasz wirtualny obraz. Bardziej złożone wykorzystanie: Możesz użyć zestawu 40 pędzli w 256 odcieniach szarości zapisanych jako IFF24 i nakładać je na 40 klatkową animację, ■ następnie po stworzeniu odbić i podświetleń symulować falujący pod wpływem wiatru ocean.

Brushmapy cechują się, podobnie jak tekstury, czterema charakterystykami: kolorem powierzchni, stopniem odbicia, stopniem przezroczystości i orientacją powierzchni. Wracając do przykładu z obrazem - użyliśmy tam prostej mapy kolorów. Mapa odbić będzie tworzyła odbicia o kolorze i natężeniu odpowiadającym mapie. Innymi słowy, czarna mapa spowoduje, że obraz nie będzie odbijający, biała mapa spowoduje, że obraz będzie miał właściwości lustra, żółta mapa będzie tworzyła refleksy tylko koloru żółtego (nie jest to całkiem prawda - będzie odbijany kolor zielony i czerwony, ponieważ z takich składowych składa się kolor żółty). Podobnie dzieje się z przezroczystością (transparency). Czarny daje nieprzezroczystość, ■ biały całkowitą przezroczystość, żółty powoduje przechodzenie światła żółtego. Zauważ, że obiekty przezroczyste mają współczynnik załamania ustawiany przez requester z atrybutami.

Ostatnim rodzajem pędzla jest wysokość. Mapa wysokości (altitude map) mówi Imagine'owi, że światło padające na powierzchnię obiektu powinno być odbite, załamane i powodować odblaski (!), tak jakby padało na powierzchnię o pewnym kształcie opisującym natężenie. Jeśli nałożyłeś rysunek z dużą ilością szarych kropek na kulę, powinieneś otrzymać odbicia i odblaski, jakby kula miała małe wypustki (coś na wzór pomarańczy). Mapa wysokości wcale nie zmienia rzeczywistej wysokości powierzchni twojego obiektu.

Opcją, której powinieneś używać przy wykorzystywaniu mapy przezroczystości jest "Full Scale Value" w requesterze pędzli. Pozwala to ■ określenie przez program, jak przezroczysty powinien być zupełnie biały obraz. Jeśli chcesz, aby czysta biel była zupełnie przezroczysta (niewidoczna) ustaw tę wartość na 255 (maksimum). Jeśli ustawisz niższą wartość, zmniejszysz przezroczystość. Jeśli tworzysz szybę w oknie, możesz ustawić wartość 200, aby ludzie widzieli, że tam jest szyba.

Jako mapa może być użyty dowolny rysunek IFF pochodzący chociażby ■ programu DeluxePaint. Format IFF24 może pochodzić np. z programu Art Department i wielu innych programów graficznych. Art Department jest świetnym narzędziem do manipulowania rysunkami 24-bitowymi (i innymi) - skalowania, łączenia i bezstratnej obróbki rysunków, o ile masz wystarczającą ilość pamięci (nie jest ■ jednak program malarski). Możesz używać różnych ilości kolorów i rozdzielczości, w zależności od zastosowania. Imagine może łatwo zapisywać rysunki 24-bitowe, możesz więc używać renderingów jako pędzli. Jeśli jesteś naprawdę zainteresowany robieniem wysokiej jakości brush map, powinieneś używać rysunków IFF24. Można wczytać edytowany rysunek do Art Department i przeskalować do wielkości ok. 900*600, następnie wyświetlić w overscanie jako ditherowany EHB, a następnie zapisać go. Teraz należy uruchomić Deluxe Painta i edytować rysunek. Przy tak dużym rysunku poszczególne punkty nie mają większego znaczenia, więc dobrze jest używać szczególnie duże brushe i airbrushe. Po zakończonej edycji pomniejsz rysunek do oryginalnych rozmiarów przy użyciu Art Departmenta. Jeśli już masz mapę i wiesz, co chcesz z nią zrobić, musisz ją umieścić na wybranym obiekcie. Określana jest wielkość i orientacja mapy ■ wpływ na obiekt, ■ w przypadku map wysokości zniekształcenia powierzchni.

Są trzy podstawowe typy nakładania - "płaskie" (Flat X, Flat Z), "sferyczne" (Wrap X, Wrap Z) i "walcowe" (Flat X, Wrap Z i Wrap X, Flat Z). Nakładanie płaskie będzie ignorowało nierówności powierzchni (bump) i inne własności powierzchni. Działa to trochę jak rzutnik slajdów. Nakładanie sferyczne otacza obiekt i odwzorowuje krzywiznę powierzchni. Przy nakładaniu walcowym krzywizna powierzchni jest uwzględniana tylko w jednym kierunku.

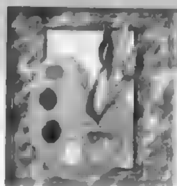


NAKŁADANIE PŁASKIE

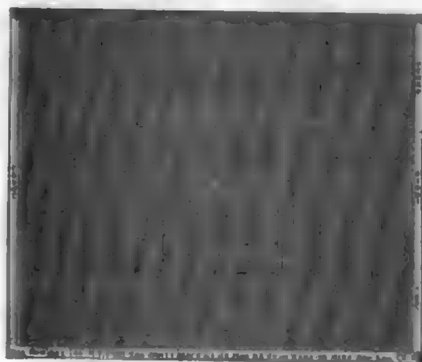
Nakładanie płaskie jest najczęściej używanym i chyba najwygodniejszym rodzajem nakładania powierzchni. Coś, jakbyśmy przyklejali plakat na ścianę. Dobrym przykładem jest napis na sześcianie.

Po pierwsze musisz mieć rysunek z napisem i sześcian. Umieszczasz wtedy rysunek na obiekcie, pewnie będzie potrzebne użycie trybu "Edit Axes". Pozwala to na poruszanie osiami myszą i tymi samymi poleceniami z klawiatury ■ przy obiektach: m - ruch, s - skalowanie, r - obrót, x, y i z do zmiany kierunku.

Niebezpieczeństwo przy edycji osi polega na tym, że chcesz być w trybie LOCAL, szczególnie przy skalowaniu osi ■ jednym kierunku (tak jakbyś chciał zwiększyć wysokość pozostawiając tę samą szerokość i głębokość). Jeśli nie chcesz skalować w trybie LOCAL, czasem twoje zmiany nie działają. Możesz to sprawdzić przez ponowne wybranie "Edit Axes" dla upewnienia się, że nic się nie zmieniło.



Oś, którą edytujesz [redacted] żółtą obwódkę, co jest bardzo mylące. Obszar, gdzie jest nakładany rysunek, to tylko górny prawy róg tej obwódki. Lewy dolny róg rysunku na prawo od środka osi, a jego górny prawy róg w punkcie zdefiniowanym przez osie X i Z.



Trzeba umieścić osie tak, żeby górny prawy fragment leżał dokładnie tam, gdzie chcesz umieścić rysunek. Jeśli chcesz, aby rysunek pokrył cały bok sześcianu, rysunek powinien być o kilka pikseli większy, tak aby przypadkiem nie powstała wokół niego obwódka.

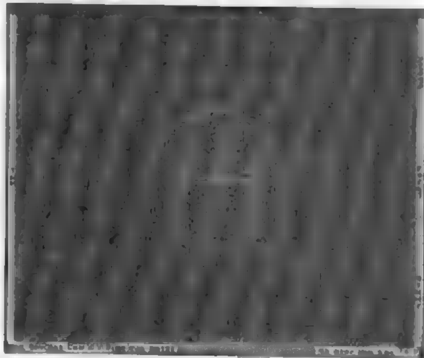
Oś Y rysunku jest również ważna. Mówi ona Imagine'owi jak głęboko użyć rysunku. Po prostu, każda część obiektu leżąca między początkiem osi, a wierzchołkiem osi Y będzie zamalowana (albo odbijająca, itp). Dla sześcianu, chciałbyś przesunąć i przeskalować osie rysunku w osi Y tak, aby oś Y przecinała jedną stronę sześcianu, ale nie drugą. Przecięta strona sześcianu powinna być pod wpływem rysunku, natomiast przeciwna strona powinna pozostać pusta. Jeśli przeskalowałeś oś Y tak, aby zawierała obie strony sześcianu, rysunek powinien znaleźć się po jej obu stronach (będziesz widział napis po drugiej stronie w odbiciu lustrzanym, ponieważ będziesz oglądał go z przeciwnej strony).

NAKŁADANIE WRAP X, WRAP Z

Są to najbardziej skomplikowane rodzaje nakładań. Oś Y nie jest tu używana, ponieważ nakładanie odbywa się wokół całej powierzchni obiektu, a nie jego części.

Jeśli chciałbyś umieścić osie tak, że oś Z zajmuje całą wysokość obiektu... ich wysokość powinna być nieco większa niż obiektu. Oś X musi być rozumiana jako "promień wnikania" i powinna być nieco większa niż połowa długości obiektu. Oś umieszczamy na środku obiektu, tak aby wystawała [redacted] jego boku u dołu obiektu. Oś Z powinna przez niego przechodzić i być nieco wyższa niż wysokość obiektu. Oś X musi być nieco dłuższa niż maksymalny promień obiektu od jego środka. Oś Y nie ma znaczenia. Pozostaw ją jaka jest.

Jeżeli nałożysz rysunek składający się z siatki linii, to całość będzie wyglądała jak układ równoleżników i południków na kuli ziemskiej.



NAKŁADANIE CYLINDRYCZNE (WRAP X, FLAT Z, FLAT X, WRAP Z)

Nakładanie rysunku jest identyczne do nakładania sferycznego, jakkolwiek otrzymany wynik jest zupełnie inny. Jeśli nałożysz cylindrycznie siatkę poziomych i pionowych linii na kulę, to otrzymasz prawidłowy układ południków, ale równoleżniki będą dalej od siebie im bliżej biegunów, [redacted] powodu równoległego nakładania linii poziomych. Poza tym przy biegunie linie będą grubsze.

Ostatnia uwaga dotycząca umieszczania osi - przy nakładaniu map wysokości, oś Y jest użyta do pomiaru natężenia symulowanego efektu. Przy nakładaniu cylindrycznym i sferycznym przeskaluj oś Y. Przy modelowaniu pomarańczy długość osi powinna wynosić 1% promienia. Przy modelowaniu zerodowanej planety możesz użyć 10%. Większa wartość będzie dawała raczej głupio wyglądające refleksy.

Oś Y użyta jest również przy płaskim nakładaniu map wysokości, co może Cię ograniczać, jeśli chciałbyś nakładać pofałdowany napis na obie strony sześcianu. W tym przypadku musisz użyć dwóch rysunków. Preferowałbym użycie gadżetu numerycznego w requesterze brushy.

Jeśli już wiesz jak umieszczać poszczególne brushe, możesz pobawić się efektami specjalnymi. Brushe można mieszać jak tekstury. Na każdym obiekcie możesz umieścić do czterech brushy, które będą kolejno wykorzystane. Możesz nałożyć [redacted] przykład mapę kolorów i mapę odbić, na ten sam obiekt i [redacted] to samo miejsce.

Jestem właściwie pewien, że brushe są nakładane po teksturach. W przeciwnym razie nie mógłbyś otrzymać drewnianych słojów na twarzy.

Ciekawe są brushmapy powtarzające się. Są one układane na obiekcie jeden przy drugim. Jeśli nakładasz w taki sposób na ziemię, to brushe będą powtarzane nieskończenie wiele razy. Wielkość każdej "klepki" jest określana przez osie podobnie jak przy normalnym nakładaniu. Używać ich możesz do efektów przy robieniu chodników, ścian z cegieł, dachówek, itp.

Nakładanie powtarzających się brushmap jest takie samo, jak zwykłe nakładanie płaskie. Wielkość brusha jest określona osiami X i Z, a "głębokość wnikania" przez oś Y.

Bardzo użyteczną opcją jest "mirror". Powoduje ona, że każda kolejna "klepka" będzie odbita lustrzanie w stosunku do sąsiedniej, dzięki temu krawędzie map kolorów zawsze do siebie pasują.

Powtarzające się brushmapy nie służą tylko do pokrywania płaszczyzny wizerunkami Twojej twarzy. Można wyobrazić sobie narysowanie wysokiej jakości cegły, w wysokiej rozdzielczości, z rysami, wgłębieniami i innymi szczegółami, a następnie utworzenie z niej ściany z mnóstwem szczegółów, w przeciwieństwie do ściany [redacted] cegieł utworzonej z pomocą zwykłej tekstury.

Ostatnia cecha brusmap jest również bardzo pożyteczna. Możesz tworzyć animowane brushmapy zmieniające się co klatkę. Rysunki te nie są animbrushami zapisywanymi przez DPainta. Są to oddzielne rysunki, dzięki czemu możesz używać brushy 24-bitowych. Aby stworzyć animowanego brusha, musisz zapisać sekwencję rysunków w takim formacie:

Mypic.0001
Mypic.0002
Mypic.0003
Mypic.0004
Mypic.0005

i tak dalej w zależności od ilości rysunków. Rozszerzenie musi mieć cztery cyfry. Aby użyć tej sekwencji w animacji musisz wpisać jako nazwę brusha "Mypic", a następnie ustawić "Max sequence #" zgodnie [redacted] ilością rysunków.

To na tyle jeśli chodzi o brushmapy. Mozesz dzięki nim znacznie ulepszyć swoje obiekty, a przecież nakładanie brushmap nie jest trudne, jeśli wiesz jak umieszczać osie. Do dzieła!

PS. Serdecznie dziękuje za pomoc panu Stevenowi Worley'owi, którego teksty znalazłem [redacted] jednym z dysków PD do Imagine, [redacted] także Arturowi za pomoc przy ich tłumaczeniu. ❑

MODEM I WSKAZÓWKI

Tomasz Matyjek

W poprzednim Amigowcu przedstawiłem modem Supra [illegible] bis. Dzisiaj zostanie podanych kilka praktycznych wskazówek, jak go optymalnie ustawić. Właściwe zaprogramowanie ma decydujący wpływ na zachowanie się modemu w różnych sytuacjach. Co jednak oznacza w praktyce programowanie modemu i jak się to robi?

Otóż modem ma wbudowany specjalny rodzaj pamięci, którą użytkownik może traktować jako zespół 8 bitowych rejestrów, do których można wpisywać wartości z zakresu 0-255. Jest to pamięć nieulotna, co znaczy, że wpisane do niej dane nie znikną po wyłączeniu zasilania. Ma ona jednak pewną wadę - może być zapisana ograniczoną liczbę razy. W praktyce liczba ta wynosi od kilkudziesięciu do kilkuset tysięcy. Biorąc pod uwagę fakt, że modem programuje się stosunkowo rzadko, nie jest to powód do zmartwień, należy tylko o tym pamiętać i wystrzegać się sytuacji, w której program do obsługi modemu sam zapisuje konfigurację. Raz ustawione parametry będą aktywne po każdym włączeniu lub resecie modemu.

Dla wygody obsługi opisywany modem ma dwa zestawy takich rejestrów. O tym, który z nich jest w danej chwili aktywny, decyduje użytkownik. Aktualną konfigurację modemu można przeczytać komendą AT&V. Spowoduje ona wypisanie na ekranie 4 ciągów znaków o następujących nazwach: ACTIVE PROFILE, STORED PROFILE 0, STORED PROFILE 1, TELEPHONE NUMBERS. Ich znaczenie jest nietrudne do odgadnięcia: ACTIVE PROFILE jest to konfiguracja aktywna w danej chwili. STORED PROFILE 0 to pierwsza z konfiguracji zapamiętanych przez użytkownika. Przeważnie jest ona jednocześnie aktywną konfiguracją. STORED PROFILE 1 jest, jak łatwo się domyślić, drugą zapamiętaną konfiguracją użytkownika, natomiast TELEPHONE NUMBERS to po prostu zapisane w pamięci modemu numery telefonów (maksymalnie cztery). O tym, które z ustawień (STORED PROFILE 0 lub 1) będzie obowiązywało po włączeniu modemu, decyduje zawartość odpowiedniego rejestru. Używa się do tego instrukcji odpowiednio AT&Y0 lub AT&Y1. Do resetu modemu służą komendy ATZ0 i ATZ1. Różnica między nimi jest niewielka - ATZ0 uaktywnia pierwszą konfigurację, a ATZ1 drugą. Dla uproszczenia, zamiast ATZ0 można pisać ATZ; uwaga ta dotyczy wszystkich komend kończących się zerem. Przykładowa, dobrze sprawdzająca się w praktyce konfiguracja przedstawia się tak:

ACTIVE PROFILE: B1 E1 L1 M1 N1 P Q0 V1 W2 X4 Y0 %C1 %E0 %G1 3 &C1 &D2 &K3 &Q5 &R0 &S0 &X0 &Y0 S00:000 S01:000 S02:043 S03:013 S04:010 S05:008 S06:002 S07:060 S08:002 S09:006 S10:106 S11:095 S12:050 S18:000 S23:063 S25:005 S26:001 S36:007 S37:000 S38:003 S40:087 S46:138 S48:007 S91:009 S95:000 S109:062 S110:002

Parametry zaczynające sie od litery S oznaczają numer rejestru i jego zawartość, np. S10:106 informuje, że w rejestr 10 wpisana jest liczba 106. Zawartosc takich rejestrów można zmieniać za pomocą komendy ATSn=x, gdzie n - numer rejestru, x - wpisywana wartość. Pozostałe parametry nie oznaczają konkretnych rejestrów, ale oczywiście również decydują o zachowaniu się modemu. Są to najczęściej zmieniane w praktyce wartości. Taki sposób jest po prostu znacznie wygodniejszy, gdyż zamiast pamiętać za [illegible] który bit jest w jakimś rejestrze odpowiedzialny, wpisujemy tylko odpowiednią komendę. Na przykład, ATL0, ATL1 lub ATL2 (L - od "loudness" czyli głośność) ustawia poziom dźwięku z wewnętrznego głośnika.

Parametry zmienione w opisany wyżej sposób mają wpływ JEDYNIE na aktualną konfigurację, która po resecie modemu lub jego wyłączeniu bezpowrotnie przepadnie. Aby tak się nie stało, należy aktualną konfigurację zapamiętać jako PROFILE 0 lub PROFILE 1. Dokonuje się tego za pomocą instrukcji AT&W0 lub AT&W1 (znowu, zamiast AT&W0 można napisać AT&W).

Może się jednak zdarzyć, że błędnie zaprogramowany modem nie będzie chciał poprawnie działać. Nie trzeba wtedy wpadać w panikę, producent przewidział także taką możliwość. Używając komendy AT&Fn, gdzie n=0, 1 lub 2, przywołujemy jedną ze "statystycznie dobrych" konfiguracji ustawionych przez fabrykę. Nie da się ich zmieniać, są na stałe zapisane w pamięci ROM modemu.

Powróćmydo opisanej wyżej konfiguracji. Liczne testy praktyczne pokazały, iż jest ona dobrze dopasowana do naszych pozostawiających wiele do życzenia linii telefonicznych. Niecierpliwi właściciele modemu mogą ją po prostu "wklepać" bez jej interpretacji. Prędzej czy później każdy jednak będzie chciał zmienić niektóre parametry, dlatego poniżej przedstawiony jest krótki opis niektórych z nich. I tak po kolei:

B0, B1 - rodzaj protokołu dla prędkości poniżej 2400.

0 - CCITT v.21/v.22,

1 - Bell 103/212A Parametr ten ma obecnie minimalne znaczenie, gdyż prędkości poniżej 2400 w zasadzie się już nie spotyka.

E0, E1 - echo komend

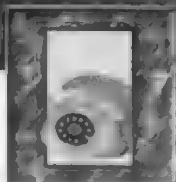
0 - wyłączone

1 - włączone

L0, L1, L2, L3 - Poziom dźwięku z wewnętrznego głośnika

0 - najniższy

3 - najwyższy



M0, M1 - praca głośnika
0 - zawsze wyłączony
1 - wyłączany po uzyskaniu połączenia

N0, N1 - dozwolona prędkość transmisji
0 - maksymalna prędkość jest ustalona w rejestrze S37
1 - połączenie odbędzie się z maksymalną możliwą prędkością

Opcja ta ma zwykle na celu wymuszenie niższej prędkości niż maksymalna dla danego modemu, gdy z góry spodziewamy się połączenia o słabej jakości.

P, T - rodzaj centrali telefonicznej
P - z wybieraniem impulsowym (pulse dial)
T - z wybieraniem tonowym (tone dial)

Q0, Q1, Q2 - aktywizacja tzw. "result codes", czyli kodów wynikowych informujących o różnych zdarzeniach (np. o uzyskaniu połączenia)
0 - włączone
1 - wyłączone
2 - włączone, gdy modem inicjuje połączenie, wyłączone, gdy odbiera

V0, V1 - rodzaj kodów wynikowych
0 - informacja tylko numeryczna
1 - informacja numeryczna i tekstowa

W0, W1, W2 - informacja o szybkości połączenia i rodzaju korekcji błędów
0 - po uzyskaniu połączenia będzie wyświetlona prędkość modem - komputer
1 - j.w. oraz rodzaj korekcji błędów
2 - po uzyskaniu połączenia będzie wyświetlona prędkość modem - modem

X0, X1, X2, X3, X4 - sposób dzwonienia
0 - modem nie czeka na sygnał ciągły z centrali, ignoruje ew. sygnał zajętości, wysyła komunikat "CONNECT" po uzyskaniu połączenia
1 - podobnie jak wyżej, ale komunikat po uzyskaniu połączenia jest typu "CONNECT xxxx", czyli zawiera informację o prędkości
2 - Modem czeka na sygnał ciągły z centrali, ignoruje ew. sygnał zajętości, wysyła komunikat "CONNECT xxxx" po uzyskaniu połączenia
3 - modem nie czeka na sygnał ciągły z centrali, wysyła komunikat "CONNECT xxxx" po uzyskaniu połączenia, oraz "BUSY", gdy sygnał jest zajęty
4 - modem czeka [redacted] sygnał ciągły z centrali, wysyła komunikat "CONNECT xxxx" po uzyskaniu połączenia, "NO DIALTONE", gdy sygnał ciągły z centrali nie pojawi się w ciągu 5 sekund, oraz "BUSY", gdy sygnał jest zajęty

Y0, Y1 - reakcja modemu po odebraniu określonej liczby znaków spacji
0 - brak reakcji
1 - przerwanie połączenia

%C0, %C1 - kompresja danych
0 - wyłączona
1 - włączona

%E0, %E1 - reakcja modemu na zmianę parametrów linii [redacted] trakcie połączenia
0 - brak reakcji
1 - inicjalizacja procesu zwanego "Retrain" Proces ten składa się z następujących faz:

a) zmiana prędkości na 4800,
b) analiza jakości linii i w miarę możliwości optymalizacja połączenia,
c) ponowne określenie możliwości obu modemów,
d) jeśli stan linii uległ poprawie, zmiana prędkości na następną możliwą - pod warunkiem, że ustawiony był parametr %G1.

%G0, %G1 - działanie podobne do %E, ale znacznie efektywniejsze podczas połączenia V32.bis.
Przy zmianie parametrów linii realizowany jest jedynie punkt d). Gdy połączenie jest typu V32, działanie tego parametru jest identyczne z %E. Oczywiste jest więc, dlaczego zalecane ustawienie to %E0 i %G1.

0, 1, 2, 3 - rodzaj połączenia
Przy 3 (zalecane), modem automatycznie wybiera najlepszy możliwy protokół transmisji.

&C0, &C1 - sposób sterowania linii DCD (Data Carrier Detect)
0 - zawsze logiczna jedynka
1 - jedynka po uzyskaniu połączenia

&D0, &D1, &D2, &D3 - reakcja modemu [redacted] zmianę stanu linii DTR (Data Terminal Ready)
0 - brak reakcji
1 - przejście do "command state" (przyjmowania komend) przy zmianie DTR z logicznego 1 [redacted] 0
2 - przerwanie aktualnego połączenia, przejście do "command state", wyłączenie opcji "auto answer" tak długo, jak sygnał DTR ma wartość logiczną 0
3 - Reset modemu przy zmianie DTR z 1 na 0

&K0, &K3, &K4, [redacted] - rodzaj potwierdzenia pakietu danych używany przez modem
0 - potwierdzenie wyłączone
3 - RTS/CTS (zdecydowanie zalecane)
4 - potwierdzenie programowe typu XON/XOFF
5 - potwierdzenie programowe typu "transparent" XON/XOFF

&R0, &R1 - rodzaj synchronizacji RTS/CTS
0 - linia CTS reaguje na linie RTS (zalecane)
1 - linia CTS ustawiana na logiczną jedynkę

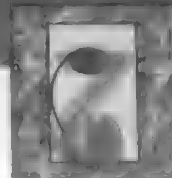
&V - wyświetlenie wszystkich konfiguracji

&X0, &X1 - zegar synchronicznej transmisji
0 - wewnętrzny (zalecane)
1 - zewnętrzny

&Y0, &Y1 - konfiguracja po włączeniu zasilania
0 - profile 0
1 - profile 1

&W0, &W1 - zapisanie aktualnej konfiguracji
0 - jako profile 0
1 - jako profile 1

Niniejszy artykuł nie ma [redacted] celu zastąpienia oryginalnej instrukcji. Mam jedynie nadzieję, że ułatwi on początkującemu użytkownikowi w miarę sprawne posługiwanie się modemem, bez konieczności mozolnego przebijania się przez cały manual. Oczywiście, po pewnym czasie sięgnięcie do instrukcji będzie zwykłą koniecznością, lecz nie będzie to już wtedy takie straszne. ❑



MAPA PROTRACKERA

[illegible]

Witam wszystkich początkujących muzyków w kolejnym odcinku naszego poradnika.

Ostatnimi czasy miałem okazję kilka razy uczestniczyć w tworzeniu tzw. "friendshipa" (czyli jednego modułu pisanego przez paru muzyków). Nie chciałbym się tutaj rozwodzić nad sensem takich przedsięwzięć, ani też opisywać walorów artystycznych powstałych w ten sposób utworów, a właściwie zwrócić uwagę na coś zupełnie innego. Otóż zauważyłem, że dla wielu muzyków (szczególnie tych początkujących) zupełnie nie istnieje klawiatura. I co więcej nie mają zielonego pojęcia o "ukrytych" mozliwościach Protrackera.

Zatem oświadczam wszystkim nieuświadomionym, że większość nader istotnych funkcji w Protrackerze jest nieosiągalna za pomocą myszy.

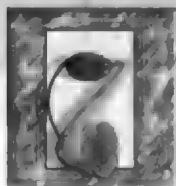
Wychodząc naprzeciw początkującym, zdecydowałem się zamieścić opis klawiatury w Protrackerze, bowiem sprawne opanowanie poniższych kombinacji niezwykle ułatwia pracę z tym programem.

Uczenie się na pamięć zazwyczaj nie daje spodziewanych rezultatów. Proponuję umieścić opis w widocznym miejscu podczas pracy np. przy monitorze i po prostu z niego korzystać.

Mapa klawiatury Protrakera w wersji 1.1MB.

Kombinacja klawiszy	Funkcja
F1	obniżenie klawiatury o jedną oktawę (od C-1 do G-3)
F2	podniesienie klawiatury o jedną oktawę (od C-2 do B-3)
F6	ustawia pozycję 00 w patternie
F7	ustawia pozycję 16 w patternie
F8	ustawia pozycję 32 w patternie
F9	ustawia pozycję 48 w patternie
F10	ustawia pozycję 63 w patternie
SHIFT+F6-F10	zapamiętuje obecną pozycję w patternie na wybranym klawiszu funkcyjnym
ALT+F6-F10	gra pattern od zapamiętanej pozycji
CTRL+F6-F10	nagrywa pattern od zapamiętanej pozycji
ESC	wyjście z większości ekranów i funkcji

Kombinacja klawiszy	Funkcja
SHIFT+RETURN	wpisuje pustą nutę na pozycji kursora w jednej ścieżce, inne nuty spycha na dół
SHIFT+BACKSPACE	kasuje nutę powyżej kursora w jednej ścieżce, inne nuty przesuwa w górę
ALT+RETURN	wpisuje pustą nutę na pozycji kursora we wszystkich ścieżkach, inne nuty spycha w dół
ALT+BACKSPACE	kasuje nutę powyżej kursora we wszystkich ścieżkah, inne nuty przesuwa w górę
CTRL+RETURN	przesuwa komendy o jeden rząd w dół
CTRL+BACKSPACE	przesuwa komendy o jeden rząd w górę
RETURN	przesuwa kursor w dół
BACKSPACE	przesuwa kursor w górę

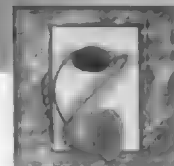


Kombinacja klawiszy	Funkcja	Kombinacja klawiszy	Funkcja
CTRL+0-9	ustawia skok pomiędzy wpisywanymi nu-	PRAWY SHIFT	włącza nagrywanie
	tami	CAPS LOCK	włącza/wyłącza keyrepeat
ALT+PRAWY/LEWY	zwiększa/zmniejsza numer patternu	DEL	kasuje tylko nutę pod kursorem
KURSOR		ALT+DEL	kasuje tylko komendę pod kursorem
SHIFT+PRAWY/LEWY	zwiększa/zmniejsza numer pozycji ut-	SHIFT+DEL	kasuje nutę i komendę pod kursorem
KURSOR	woru	SHIFT+F3	wycina ścieżkę i kopiuje do bufora
CTRL+PRAWY/LEWY	zwiększa/zmniejsza numer sampla	SHIFT+F4	kopiuje ścieżkę do bufora
KURSOR		SHIFT+F5	kopiuje bufor do ścieżki
`(klawisz nad TAB)	wychodzi do CLI	ALT+F3	wycina pattern i kopiuje do bufora
HELP	włącza ekran HELP lub PLST	ALT+F4	kopiuje pattern do bufora
SHIFT+HELP	zmienia funkcję klawisza HELP	ALT+F5	kopiuje bufor do patternu
SPACE	włącza/wyłącza tryb edycji	CTRL+F3	wycina komendy i kopiuje do bufora
PRAWA AMIGA	gra pattern	CTRL+F4	kopiuje komendy do bufora
PRAWY ALT	gra utwór	CTRL+F5	kopiuje komendy z bufora do ścieżki

LEWY KLAWISZ AMIGA + klawisze:

wybrany sampl/ścieżka		wybrany sampl/pattern	
1	nuta w górę	2	nuta w górę
[illegible]	nuta w dół	W	nuta w dół
A	oktawa w górę	S	oktawa w górę
Z	oktawa w dół	X	oktawa w dół
wszystko/ścieżka		**wszystko/pattern**	
3	nuta w górę	4	nuta w górę
E	nuta w dół	R	nuta w dół
D	oktawa w górę	F	oktawa w górę
C	oktawa w dół	V	oktawa w dół

Kombinacja klawiszy	Funkcja	Kombinacja klawiszy	Funkcja
TAB	przesuwa kursor do następnej ścieżki	ENTER	wybór sampla 10
SHIFT+TAB	przesuwa kursor do poprzedniej ścieżki	ENTER+klawiatura numeryczna	wybór sampli 11-1F
klawiatura numeryczna	wybór sampli 00-0F	.	kasuje sampla (po potwierdzeniu)



Kombinacja klawiszy	Funkcja
\	zmienia tryb keypad (możliwość podsłuchu przy wyborze sampli)
CTRL+A	włącza/wyłącza kanał
CTRL+B	zaznacza blok
CTRL+C	kopiuje blok do bufora
CTRL+D	kasuje blok i przesuwa nuty do góry
CTRL+E	rozszerza ścieżkę
CTRL+F	włącza/wyłącza filtr
CTRL+G	wzmacnia (boostuje) wszystkie sample
CTRL+H	zwiększa wysokość wszystkich nut w zaznaczonym bloku
CTRL+I	wstawia blok do ścieżki i spycha nuty w dół
CTRL+J	dołącza nowy blok do poprzednio wstawionego
CTRL+K	kasuje ścieżkę od pozycji kursora do końca ścieżki
CTRL+L	zmniejsza wysokość wszystkich nut w bloku
CTRL+M	włącza/wyłącza multiklawiaturę
CTRL+N	ponownie zaznacza ostatni blok
CTRL+O	zmniejsza ścieżkę
CTRL+P	wkleja blok
CTRL+R	przywraca pozycje F6-F10
CTRL+S	włącza/wyłącza split klawiaturę
CTRL+T	wymienia ścieżki między sobą
CTRL+U	kasuje ostatnią zmianę (UNDO)
CTRL+V	filtruje wszystkie sample
CTRL+W	polifonizuje blok
CTRL+X	wycina blok i kopiuje go do bufora
CTRL+Y	cofa blok

Kombinacja klawiszy	Funkcja
CTRL+Z	przywraca komendy
SHIFT+0-9	przyporządkowuje obecną komendę wybranemu klawiszowi
ALT+0-9	wpisuje komendę do ścieżki
ALT+""	wpisuje komendę nad kursorem do obecnej pozycji w patternie
ALT+"="	wpisuje komendę nad kursorem do obecnej pozycji w patternie, dodaje 1 do wartości komendy
ALT+"-"	wpisuje komendę nad kursorem do obecnej pozycji w patternie, odejmuje 1 do wartości komendy
ALT+A	monitor - podsłuch samplera
ALT+B	monitor - podsłuch samplera - wzmacnia (boostuje) sampla
ALT+C	włącza/wyłącza kanał 3
ALT+D	DISK OPERATION
ALT+E	EDIT OPTIONS
ALT+F	filtruje sampla
ALT+I	nastawia głośność
ALT+K	nastawia głośność - kasuje obecny sampl/ścieżkę
ALT+Q	wyjście z Protrackera
ALT+R	resampluje sampla
ALT+S	SAMPLER
ALT+T	generuje strojący dźwięk
ALT+V	włącza/wyłącza kanał 4
ALT+X	włącza/wyłącza kanał 2
ALT+Z	włącza/wyłącza kanał 1

Tak oto wygląda mapa klawiatury Protrackera w wersji 1.1B (opracowana na podstawie pliku PT.HELP). W następnym numerze zaprezentuję niezwykle przydatną w pracy tabelę akordów oraz (o ile starczy miejsca) parę praktycznych przykładów.

SPOKOJNE ŻYCIE

Nikt z nas chyba nie lubi bawić się w dużych ilościach w dynamiczne przydzielanie pamięci czy operacje na plikach. Są to jednak najbardziej podstawowe czynności wykonywane przez większość programów.

Jak mówi amerykańskie przysłowie: "Jeśli nie możesz pokonać nieprzyjaciela, to się do niego przyłącz", a ułatwienie roboty (przede wszystkim sobie) to przecież specjalność ludzi od komputerów...

Przydzielanie pamięci poprzez system odbywa się przy pomocy funkcji AllocMem() deklarowanej następująco:

void *AllocMem(ULONG size, ULONG flags)

gdzie parametr size określa wielkość przydzielanego obszaru pamięci w bajtach, natomiast flags określa typ pamięci. Dozwolone są następujące bity flags:

- MEMF_ANY (typ pamięci nie gra roli),
- MEMF_CHIP (przydzielona zostanie pamięć typu chip),
- MEMF_FAST (przydzielona zostanie pamięć typu fast),
- MEMF_CLEAR (przydzielony obszar pamięci zostanie wyzerowany),
- MEMF_PUBLIC (przydzielona pamięć może zostać wykorzystana przez inne taski lub przerwania),
- MEMF_LOCAL (pamięć zostaje zachowana przy resecie),
- MEMF_24BITDMA (pamięć z dostępem DMA adresowanym 24 bitami),
- MEMF_REVERSE (przydzielenie następuje od góry do dołu).

Zwracany przez funkcję wskaźnik to adres początku przydzielonego obszaru pamięci lub NULL w przypadku błędu. Zwolnienie pamięci odbywa się poprzez wywołanie funkcji Free Mem() następującej postaci:

void FreeMem(void *mem, ULONG size)

Pierwszy parametr to adres początku obszaru pamięci, drugi to wielkość zwalnianego obszaru. I tu zwykle leży pies pogrzebany. Jeśli przydzielamy pamięć, musimy pamiętać, jak duży obszar został przez nas zajęty. Jeśli jest to wielkość stała w programie, zwykle nie ma kłopotu, kiedy jednak przydzielamy obszary różnej wielkości (np. w zależności od wielkości wczytywanych plików), wtedy musimy powoływać do życia tablice danych i tym podobne mało eleganckie rzeczy. Należy również pamiętać, by nie zwalniać jednego obszaru kilka razy, bo GURU pewne. Zwykle można zabezpieczyć się poprzez sprawdzenie, czy wskaźnik jest różny od zera, lecz unikniemy w ten sposób tylko jednej przykrej sytuacji. Druga spotka nas, gdy wskaźnik będzie różny od zera, a funkcja FreeMem() już była wywołana...

W języku C++ wprowadzono dwa operatory dynamicznego zarządzania pamięcią: new i delete. W przypadku new wystarczy podać wielkość przydzielanego obszaru, by otrzymać wskaźnik na ten obszar. Zwolnienie pamięci odbywa się poprzez wywołanie delete z jednym tylko argumentem - adresem początku zwalnianego obszaru. Ponieważ ciągle brak na Amigę sprawdzonych, dobrych kompilatorów C++ (wielka szkoda, niestety tu Amiga jest daleko za murzynami, choć SAS C zmierza w dobrą stronę) spróbujemy przenieść ten pomysł na grunt zwykłego C.

Gdzie można by przechować informację o wielkości przydzielonego obszaru pamięci? Chyba najprościej w tym obszarze. Wystarczy, że nowa funkcja:

UBYTE *New(ULONG size, ULONG type)

przydzieli obszar pamięci typu type (wartości jak parametr flags w AllocMem()) o wielkości (size + 4) bajtów. Pierwsze 4 bajty posłużą nam za miejsce przechowania danej typu ULONG zawierającej informację o wielkości przydzielonego obszaru. By nie zniszczyć tej danej, funkcja powinna zwrócić adres przydzielonej pamięci zwiększony o 4 bajty. Cały proces można zrealizować następująco:

```
UBYTE *New( ULONG size, ULONG type )
{
    UBYTE *pointer;

    if ( !( pointer = (UBYTE *) AllocMem( size + 4, type ) ) )
      ErrorOccured( NULL, IOE_NO_MEMORY );

    *( (ULONG *) pointer ) = size + 4;
    return( pointer + 4 );
}
```

W przypadku wystąpienia błędu funkcja automatycznie wywołuje funkcję obsługi błędów o nazwie ErrorOccured() (funkcją tą zajmiemy się za chwilę).

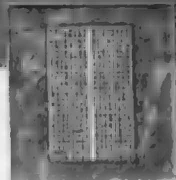
Usunięcie pamięci można zrealizować następującą funkcją:

void Delete(UBYTE **pointer)

Pierwszą rzeczą, na jaką warto zwrócić uwagę jest to, że argumentem funkcji jest wskaźnik do wskaźnika na początek obszaru. Otóż po zwolnieniu pamięci można dzięki temu wyzerować wskaźnik. W takim przypadku, jeśli tylko podczas uruchamiania programu pamiętało się o wyzerowaniu wskaźnika, będzie on równy zero zawsze wtedy, gdy nie został wykorzystany, a różny od zera, gdy wskazuje na jakiś fragment pamięci. Samo zwalnianie pamięci jest proste: wystarczy od otrzymanego adresu odjąć 4 bajty by uzyskać wskaźnik na zmienną ULONG przechowującą informację o wielkości obszaru. Wskaźnik ten jest jednocześnie wskaźnikiem na początek zwalnianej pamięci, stąd używany jest jako jeden z parametrów funkcji FreeMem(). Zwróćmy uwagę na fakt, że 4 pierwsze bajty pamięci muszą zawierać wielkość całego przydzielanego/zwalnianego fragmentu pamięci wraz z tymi bajtami! Całą procedurę można napisać następująco:

```
void Delete( UBYTE **pointer )
{
    UBYTE *wskaznik;

    if ( ( wskaznik = *pointer ) )
      {
        wskaznik -= 4;
        FreeMem( wskaznik, *( (ULONG *) wskaznik ) );
        *pointer = NULL;
      }
}
```



Zwolnienie pamięci jest realizowane jedynie wtedy, gdy wskaźnik pointer jest różny od zera. Dzięki temu można pamięć zwalniać "wielokrotnie" (tzn. można kilka razy wywołać funkcję Delete() dla danego wskaźnika i nie spowodujemy tym samym zawieszenia systemu, bowiem zwolnienie nastąpiło tylko przy pierwszym wywołaniu, potem wskaźnik jest już równy NULL).

Jak widać, napisanie krótkich funkcji może pociągnąć za sobą znaczne ułatwienie życia. Poułatwiajmy je sobie dalej i zobaczmy, co można zrobić z tak mało atrakcyjnymi funkcjami jak fopen(), fclose(), fseek(), fread() i fwrite().

Funkcja fopen() jest deklarowana jako:

FILE *fopen(const char *filename, const char *mode)

Pierwszy parametr to nazwa pliku, natomiast drugi określa sposób otwarcia pliku. Dozwolone parametry są następujące:

- "a" - otwiera plik do rozszerzania. Jeśli plik istnieje, przyjęta pozycja w pliku jest przesunięta o jeden znak za ostatni bajt pliku. Jeśli plik nie istnieje, jest stwarzany. W obu przypadkach plik jest otwierany jako write-only.
- "ab" - jak "a", lecz otwierany strumień jest binarny.
- "a+" - jak "a", lecz plik może być również odczytywany (read).
- "ab+" - jak "a+", lecz strumień jest otwierany jako binarny.
- "r" - otwiera plik do wprowadzania (read). Jeśli plik jest otwarty, wtedy pozycja w pliku jest ustawiana na pierwszy bajt pliku. Jeśli plik nie istnieje, wtedy funkcja fopen() zwraca wartość NULL.
- "rb" - jak "r", lecz otwierany strumień jest binarny.
- "r+" - otwiera plik do aktualizacji (read, write).
- "rb+" - jak "r+", lecz strumień jest otwierany jako binarny.
- "w" - otwiera plik do wyprowadzania (write). Jeśli plik istnieje, jego wielkość staje się równa 0. Jeśli plik nie istnieje, jest tworzony (również w tym przypadku długość pliku wynosi 0).
- "wb" - jak "w", lecz otwierany strumień jest binarny.
- "w+" - jak "w", lecz plik może być również odczytywany (read).
- "wb+" - jak "w+", lecz strumień jest otwierany jako binarny.

Zwracana przez funkcję fopen() wartość to wskaźnik na otwarty plik (konkretnie: strumień). Zamykanie strumienia realizuje funkcja fclose():

int fclose(FILE *stream)

której jedynym parametrem jest wskaźnik na strumień. Zwracana wartość to 0, gdy wszystko jest OK lub "nie-zero" w przypadku błędu.

Jak zwykle dążymy do maksymalnego uproszczenia zapisu (i życia), stąd warto napisać funkcję otwierającą plik, która automatycznie sprawdzi, czy urządzenie (np. dyskietka) do którego się odwołujemy istnieje w systemie, a następnie otworzy plik lub w przypadku błędu wywoła standardową procedurę obsługi błędów. Można to zrobić np. tak:

```
FILE *OpenFile( STRPTR name, STRPTR mode )
{
    FILE *file;

    TestIfVolumeExists( name );

    if ( !( file = fopen( (char *) name, (char *) mode ) ) )
      ErrorOccured( name, IOE_NO_FILE );

    return( file );
}
```

W ciele funkcji znajduje się znana nam już, choć ciągle tajemnicza ErrorOccured() oraz wywołanie procedury o nazwie TestIfVolumeExists(). Funkcji tej chciałbym poświęcić kilka zdań wyjaśnienia. Jej argumentem jest ciąg tekstowy zawierający nazwę pliku wraz z urządzeniem (cała ścieżka dostępu). Otóż funkcja TestIfVolumeExists() analizuje ten ciąg i odszukuje znak ':' (dwukropek). Znaki przed dwukropkiem traktowane są jako nazwa urządzenia, na którym znajduje się plik. Funkcja przystępuje następnie do sprawdzenia, czy takie urządzenie jest obecne w systemie, jeśli nie, wywołuje ErrorOccured(). Sposób odszukiwania urządzeń został przeze mnie omówiony w artykule "Przyjaciółko, pokaż co masz" w jednym z wcześniejszych numerów Amigowca, stąd ominę tu szczegółowy opis.

Wróćmy do zabawy z plikami. Pozostaje napisać procedurę zamykającą plik. Tu również warto sprawdzać, czy wskaźnik na zamykany plik jest różny od zera. Jeśli nie, wtedy należy zamknąć plik i wyzerować wskaźnik. Dzięki temu zabiegowi możemy - analogicznie jak przy funkcji Delete() - wywoływać funkcję wielokrotnie bez narażania się na kłopoty. Funkcja ta, nazwana przeze mnie CloseFile(), jest następującej postaci:

```
void CloseFile( FILE **file )
{
    if ( *file )
      fclose( *file );

    *file = NULL;
}
```

Do ustawiania pozycji w pliku wykorzystywana jest funkcja

int fseek(FILE *stream, long offset, int origin)

Pierwszy argument to wskaźnik na otwarty plik, offset to wartość przesunięcia w bajtach, natomiast parametr origin opisuje, od jakiego punktu należy to przesunięcie mierzyć:

- SEEK_SET - przesunięcie względem początku pliku,
- SEEK_CUR - przesunięcie względem aktualnej pozycji w pliku,
- SEEK_END - przesunięcie względem końca pliku.

Nasza wersja tej funkcji może wyglądać następująco:

```
void SetPosInFile( STRPTR name, FILE *file, ULONG position,
                                                UWORD where )
{
  if ( fseek( file, position, where ) )
    ErrorOccured( name, IOE_FILE_ERROR );
}
```

Parametr name zawierający nazwę pliku związany jest z tym, że funkcja ErrorOccured() wyprowadza komunikat, w którym podaje, podczas operacji, z jakim plikiem wystąpiły kłopoty.

Skoro operacje otwierania i zamykania plików oraz ustawiania określonej pozycji w pliku mamy zrobione, pozostaje "szczegół" w postaci funkcji wczytującej i zapisującej dane. Funkcje to fread() oraz fwrite(). Ich deklaracje wyglądają następująco (typ size_t definiowany jest jako unsigned long):

size_t fread(void *buffer, size_t size, size_t count, FILE *stream)

size_t fwrite(const void *buffer, size_t size, size_t count, FILE *stream)

W obu przypadkach znaczenie parametrów jest podobne: buffer to wskaźnik początku wczytywanego (fread()) lub zapisywanego (fwrite()) obszaru pamięci, size to ilość wczytywanych/zapisywanych bajtów, count - liczba obiektów (krotność operacji) oraz wskaźnik na otwarty plik stream. Obie funkcje zwracają parametr count w przypadku poprawnego wykonania operacji. Ponieważ chcemy by nasze funkcje zapisywały lub odczytywały pojedyncze bloki pamięci, więc funkcje fread() i fwrite() wykorzystamy następująco:

```
void ReadFromFile( STRPTR name, FILE *file, UBYTE *data, ULONG size )
{
  if ( fread( data, size, 1, file ) != 1 )
    ErrorOccured( name, IOE_FIND_ERROR );
}

void WriteToFile( STRPTR name, FILE *file, UBYTE *data, ULONG size )
{
  if ( fwrite( data, size, 1, file ) != 1 )
    ErrorOccured( name, IOE_FIND_ERROR );
}
```

Skoro już wybrzydzam, warto napisać dwie "super-funkcje", czyli po prostu funkcje, które oprócz operacji odczytywania i zapisywania danych otwierają i zamykają odpowiednie strumienie. Funkcje takie mogą się przydać do pojedynczych operacji na plikach. I tak, pierwsza funkcja ma postać:

```
void ReadFile( STRPTR name, LONG position, UBYTE *memory,
                                        ULONG length )
{
  FILE *file;
```

```
  TestIfVolumeExists( name );
  file = OpenFile( name, (STRPTR) "r" );

  if ( position )
    SetPosInFile( name, file, position, SEEK_SET );

  ReadFromFile( name, file, memory, length );
  fclose( file );
  return;
}
```

Parameter name to nazwa pliku wraz ze ścieżką dostępu, position określa pozycję w pliku, memory wskazuje na początek obszaru pamięci, w który dane mają zostać wczytane, natomiast length to ilość bajtów do wczytania. Analogicznie zbudować można funkcję zapisującą dane:

```
void SaveFile( STRPTR name, UBYTE *memory, ULONG length )
{
  FILE *file;

  TestIfVolumeExists( name );
  file = OpenFile( name, (STRPTR) "w" );
  WriteToFile( name, file, memory, length );
  fclose( [redacted] );
}
```

To tyle jeśli chodzi o operacje na plikach. Pozostaje jeszcze bliżej nie znana funkcja ErrorOccured(). Funkcja ta zwykle różni się w poszczególnych programach, inny bowiem może być sposób wyprowadzania komunikatów o błędach (np. na standardowy I/O, na rastport itp.). Konstrukcja tej funkcji może być mniej więcej następująca:

```
void ErrorOccured( STRPTR file, UBYTE error )
{
  UBYTE loop;
  LONG byk;

  if ( error == IOE_FIND_ERROR )
    {
      byk = IoErr();

      switch ( error )
        {
          case 205:
            error = IOE_NO_FILE;
            break;

          [redacted] 214:
            error = IOE_WRITE_PROTECT;
            break;

          default:
            error = IOE_FILE_ERROR;
            break;
        }
    }

  switch ( error )
    {
      case IOE_NO_MEMORY:
        printf( "Błąd: brak pamięci!" );
        break;

      case IOE_NO_FILE:
        printf( "Błąd: nie można znaleźć pliku %s", file );
        break;

      case IOE_WRITE_PROTECT:
        printf( "Błąd: Zapis pliku %s na zabezpieczony dysk!", file );
        break;

      case IOE_FILE_ERROR:
        printf( "Błąd: Nieudana operacja na pliku %s!", file );
        break;
    }

  exit(0);
}
```

Zasada działania funkcji jest prosta: jeśli error wykazuje błąd braku pamięci, wyświetl komunikat o jej braku i zakończ program. Jeśli error wykazuje wartość makra IOE_FIND_ERROR, to należy sprawdzić, jaki błąd wystąpił podczas operacji [redacted] pliku (funkcja IoErr()) i odpowiednio go sklasyfikuj. Pozostałe wartości parametru error muszą odpowiadać wartościom makr IOE_... (nie muszę chyba dodawać, że makra IOE_... definiujemy sami).

Dołączony listing zawiera omówiony zestaw funkcji oraz krótki program demonstrujący ich zastosowanie. Sam program najlepiej uruchomić z shella. Procedura TestIfVolumeExists() sprawdza tutaj, czy istnieje określone urządzenie, jeśli nie, drukuje tekst o błędzie oraz odczekuje 1 sekundę, zanim ponownie rozpocznie przeszukiwanie listy. Ponieważ w przykładzie odwołujemy się do urządzenia TEST:, należy w momencie pojawienia się komunikatu o błędzie wykonać polecenie:

assign TEST: ram:

wykorzystując funkcję Execute Command... z menu Workbench na Workbenchu.

Podany zestaw funkcji to tylko propozycja - zarys pewnego sposobu postępowania. Ostateczny kształt funkcji oraz ich ilość zależą jedynie od potrzeb i inwencji Czytelnika. ❑

```
#include <clib/exec_protos.h>
#include <clib/dos_protos.h>
#include <libraries/dosextens.h>
#include <exec/memory.h>
#include <stdlib.h>
#include <stdio.h>

extern struct DosLibrary *DOSBase;

/* Makra opisujące błędy. */

#define IOE_NO_MEMORY       0
#define IOE_NO_FILE         1
#define IOE_FILE_ERROR      2
#define IOE_FIND_ERROR      3
#define IOE_WRITE_PROTECT   4

/* Deklaracje funkcji. */

UBYTE *New( ULONG size, ULONG type );
void Delete( UBYTE **pointer );
FILE *OpenFile( STRPTR name, STRPTR mode );
void CloseFile( FILE **file );
void SetPosInFile( STRPTR name, FILE *file, ULONG position,
                                                   UWORD where );
void ReadFromFile( STRPTR name, FILE *file, UBYTE *data, ULONG size );
void WriteToFile( STRPTR name, FILE *file, UBYTE *data, ULONG size );
void ReadFile( STRPTR name, LONG position, UBYTE *memory,
                                                   ULONG length );
void SaveFile( STRPTR name, UBYTE *memory, ULONG length );
void TestIfVolumeExists( STRPTR name );
void ErrorOccured( STRPTR file, UBYTE error );

/* Definicje funkcji. */



/* Funkcja przydziela pamięć typu type o wielkości size i zwraca wskaźnik na   */
/* przydzielony obszar. Faktycznie przydzielany jest obszar o 4 bajty          */
/* większy. Bajty te zawierają rozmiar przydzielanej pamięci. Zwracany adres   */
/* wskazuje więc pierwszy bajt po 4 bajtach rozmiaru. W przypadku błędu        */
/* wywoływana jest standardowa procedura obsługi błędów.                       */

UBYTE *New( ULONG size, ULONG type )
{
  UBYTE *pointer;

    if ( !( pointer = (UBYTE *) AllocMem( size + 4, type ) ) )
      ErrorOccured( NULL, IOE_NO_MEMORY );

    *( (ULONG *) pointer ) = size + 4;
```



```
    return( pointer + 4 );
}
/* Funkcja usuwa obszar pamięci, [redacted] który wskazuje wskaźnik */
/* wskazywany przez pointer. Ponieważ 4 bajty przed adresem */
/* przydzielonego obszaru pamięci jest zmienna typu ULONG */
/* zawierająca wielkość zwalnianego obszaru pamięci, więc    */
/* wskaźnik należy przesunąć w tył o cztery bajty. Wskaźnik  */
/* na zwalniany obszar jest zerowany.                        */

void Delete( UBYTE **pointer )
{
  UBYTE *wskaznik;

    if ( ( wskaznik = *pointer ) )
      {
        wskaznik -= 4;
        FreeMem( wskaznik, *( (ULONG *) wskaznik ) );
        *pointer = NULL;
      }
}


/* Funkcja otwiera plik o nazwie name do operacji określonych przez mode */
/* ("r", "w", "a", "r+", "w+", "a+").                                   */

FILE *OpenFile( STRPTR name, STRPTR mode )
{
  FILE *file;

  TestIfVolumeExists( name );

  if ( !( file = fopen( (char *) name, (char *) mode ) ) )
    ErrorOccured( name, IOE_NO_FILE );

  return( file );
}


/* Funkcja zamyka otwarty plik wskazywany przez file oraz zeruje wskaźnik. */

void CloseFile( FILE **file )
{
  if ( *file )
    fclose( *file );

  *file = NULL;
}


/* Funkcja ustawia kursor w pliku file o nazwie name na position bajtów */
/* od miejsca określonego przez where (SEEK_SET, SEEK_CUR,              */
/* SEEK_END).                                                           */

void SetPosInFile( STRPTR name, FILE *file, ULONG position,
                                                  UWORD where )
{
  if ( fseek( file, position, where ) )
    ErrorOccured( name, IOE_FILE_ERROR );
}


/* Funkcja odczytuje z pliku file o nazwie name size bajtów w obszar */
/* wskazywany przez data.                                            */

void ReadFromFile( STRPTR name, FILE *file, UBYTE *data, ULONG size )
{
  if ( fread( data, size, 1, file ) != 1 )
    ErrorOccured( name, IOE_FIND_ERROR );
}


/* Funkcja zapisuje do pliku file o nazwie name dane wielkości size */
/* wskazywane przez data.                                           */

void WriteToFile( STRPTR name, FILE *file, UBYTE *data, ULONG size )
{
  if ( fwrite( data, size, 1, file ) != 1 )
    ErrorOccured( name, IOE_FIND_ERROR );
}


/* Funkcja otwiera plik o nazwie name i wczytuje ilość bajtów równą length  */
/* z pozycji position w obszar pamięci memory. Funkcja nie przydziela      */
/* pamięci. Gdy position jest równe 0 odczyt następuje od początku pliku.  */

void ReadFile( STRPTR name, LONG position, UBYTE *memory,
                                                  ULONG length )
{
  FILE *file;

  TestIfVolumeExists( name );
  file = OpenFile( name, (STRPTR) "r" );

  if ( position )
    SetPosInFile( name, file, position, SEEK_SET );

  ReadFromFile( [redacted] file, memory, length );
  fclose( file );
  return;
}


/* Funkcja otwiera plik o nazwie name i zapisuje w nim obszar pamięci */
/* zaczynający się [redacted] memory o wielkości length bajtów.       */

void SaveFile( STRPTR name, UBYTE *memory, ULONG length )
{
  FILE *file;

  TestIfVolumeExists( name );
  file = OpenFile( name, (STRPTR) "w" );
  WriteToFile( name, file, memory, length );
  fclose( file );
}


/* Funkcja sprawdza, czy istnieje w systemie urządzenie logiczne, którego */
/* nazwa znajduje się na początku ciągu name.                             */

void TestIfVolumeExists( STRPTR name )
{
  BOOL volume_exists;
  UBYTE loop, pos, znak1, znak2;
  ULONG brud;
  struct RootNode *rootnode;
  struct DosInfo *dosinfo;
  struct DeviceList *pointer;

  pos = 0;

  /* Odszukanie dwukropka w ciągu i zastąpienie go zerem. */

  while ( *(name + pos) != ':' )
    pos++;

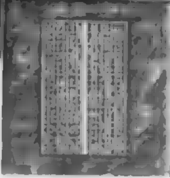
  *(name + pos) = 0;

  /* Rozpoczęcie przeszukiwania listy. */

  while(1)
    {
      Disable();
      rootnode = (struct RootNode *) DOSBase->dl_Root;
      brud = ( (ULONG) rootnode->rn_Info ) << 2;
      dosinfo = (struct DosInfo *) brud;
      brud = ( (ULONG) dosinfo->di_DevInfo ) << 2;
      pointer = (struct DeviceList *) brud;
      Enable();

      while( pointer )
        {
          volume_exists = TRUE;
          brud = ( (ULONG) ( pointer->dl_Name ) ) << 2;

          /* Czy nazwy poszukiwanego urządzenia i aktualnego w liście */
          /* są  identyczne?                                          */
```



```
        for ( loop=0; loop<=pos; loop++ )
            {
            znak1 = *( (STRPTR) (brud + loop + 1) );
            znak2 = *(name + loop);

            if ( znak1 > 96 && znak1 < 123 )
                znak1 -= 32;

            if ( znak2 > 96 && znak2 < 123 )
                znak2 -= 32;

            if ( znak1 != znak2 )
                {
                volume_exists = FALSE;
                break;
                }
            }

        if ( volume_exists )
            if ( pointer->dl_Task )
                {
                /* Nazwy są identyczne, urządzenie jest w systemie.  */
                /* W miejsce znaku zero wstaw znaczący dwukropek. */

                *(name + pos) = ':';
                return;
                }

        brud = ( (ULONG) ( pointer->dl_Next ) ) << 2;
        pointer = (struct DeviceList *) brud;
        }

    printf( "Brak urządzenia %s!!!", name );
    Delay(50);
    }
}


/* Funkcja obsługująca błędy. */

void ErrorOccured( STRPTR file, UBYTE error )
{
  LONG byk;

  if ( error == IOE_FIND_ERROR )
    {
      byk = IoErr();

      switch ( error )
        {
          case 205:
            error = IOE_NO_FILE;
            break;

          case 214:
            error = IOE_WRITE_PROTECT;
            break;

          default:
            error = IOE_FILE_ERROR;
            break;
        }
    }

  switch ( error )
    {
      case IOE_NO_MEMORY:
        printf( "Błąd: brak pamięci!" );
        break;

      case IOE_NO_FILE:
        printf( "Błąd: nie można znaleźć pliku %s", file );
        break;

      case IOE_WRITE_PROTECT:
        printf( "Błąd. Zapis pliku %s na zabezpieczony dysk!", file );
        break;

      case IOE_FILE_ERROR:
        printf( "Błąd. Nieudana operacja na pliku %s!", file );
          break;
    }

  exit(0);
}


/* Tablica ze znakami. */

UBYTE Tekst[30] = "CHRZANIĘ   RYCHA   KOWALSKIEGO!!";


/* Program główny. */

main()
{
  UBYTE loop;
  UBYTE *bufor;
  FILE *file;

  /* Przydzielenie bufora. */

  bufor = New( 30, MEMF_ANY | MEMF_CLEAR );

  /* Przekopiowanie tekstu z tablicy do bufora. */

  for ( loop=0; loop<30; loop++ )
    *( bufor + loop ) = Tekst[loop];

  /* Zgranie zawartości bufora do pliku test w urządzeniu TEST:        */
  /* Tego urządzenia nie powinno być wcześniej w systemie, stąd        */
  /* program będzie oczekiwał wewnątrz funkcji TestIfVolumeExists()   */
  /* na wprowadzenie z Workbencha polecenia                            */
  /*      assign TEST: ram:                                             */

  SaveFile( "TEST:test", bufor, 30 );

  /* Skasowanie bufora - argumentem jest adres do wskaźnika!. */

  Delete( &bufor );

  /* Wyczyszczenie tablicy. */

  for ( loop=0; loop<30; loop++ )
    Tekst[loop] = 0;

  printf( "Test 1: %s", Tekst );

  /* Wczytanie danych z pliku. */

  ReadFile( "TEST:test", 0, &Tekst[0], 30 );
  printf( "Test 2: %s", Tekst );

  /* Ponawiamy próbę skasowania bufora - nic powinno się nic stać. */

  Delete( &bufor );

  /* Otwarcie pliku. */

  file = OpenFile( "TEST:test", "r+" );

  /* Przesunięcie na 10 pozycję w pliku. */

  SetPosInFile( "TEST:test", file, 10, SEEK_SET );

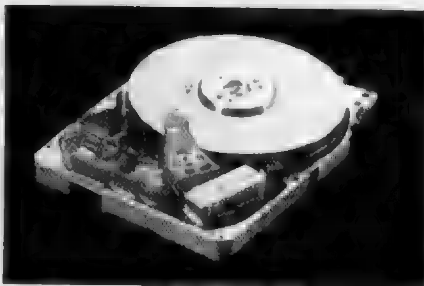
  /* Próba wczytania 30 bajtów. */

  ReadFromFile( "TEST:test", file, &Tekst[0], 30 );

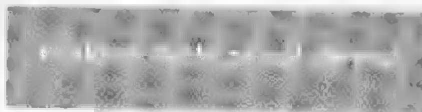
  /* Ponieważ instrukcja poprzednia powoduje wczytanie 30 bajtów z pliku   */
  /* o długości 30 bajtów, lecz poczynając od bajtu 10, więc wystąpi błąd. */
  /* Zostanie wywołana funkcja ErrorOccured(), która zakończy program,     */
  /* stąd nie uda nam się, niestety, wykonać poniższej instrukcji          */
  /* zamknięcia pliku. By zmazać taki niezamknięty plik najlepiej wykonać  */
  /* tzw. "Trzech Króli", czyli reset.                                     */

  /* Zamknięcie pliku - argumentem jest adres do pliku.                    */

  CloseFile( &file );
}
```



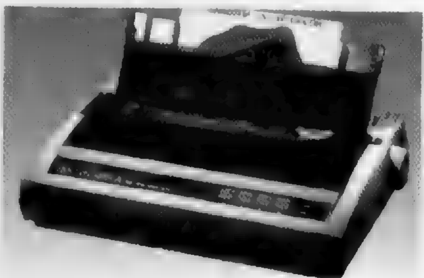
Producent: Stefan Ossowski's Schatztruhe
Dystrybutor: Stefan Ossowski's Schatztruhe Veronikastraße 33, 45131 Essen, RFN
Zgodność: 1.2, 1.3, 2.x, 3.x
Cena: 60-80DM



DiskExpander jest programikiem, dzięki któremu będziecie mogli powiększyć prawie ■ połowę objętość dyskietek i dysków twardych. Całe jego działanie sprowadza się pakowania danych od 30% do 70% ich do pierwotnej wielkości. Przy czym do dyspozycji stoją różnego rodzaju algorytmy. Jedne są wydajniejsze, ale wolniejsze, inne zaś pakują gorzej, ale za to ich szybkość jest znośna. Program nie tylko umożliwia władowanie na normalną dyskietkę 1.5 MB danych, ale również zwiększa podwójnie zawartość RADu. Również dysk twardy może zostać spakowany. Owszem będzie trochę wolniej chodził, ale przy pojemności 20MB da się na nim trochę więcej upchać. Program działa na wszystkich Amigach wyposażonych w min. 1 MB pamięci.

Prócz programu podstawowego na dysku znajduje się program "DevicePacker" umożliwiający pakowanie osobnych urządzeń, czy choćby tylko katalogów oraz programik DEStatisik, który pozwala cieszyć się zyskami przestrzeni dyskowej stworzonymi przez DiskExpandera.

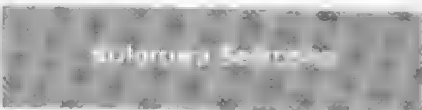
Program dostarczany jest w wersji niemieckiej. Oryginalna dyskietka jest zmyślnie zabezpieczona, ■ program działa dopiero po jego prawidłowym zainstalowaniu. Producent ostrzega przed używaniem nielegalnych kopii, ponieważ może to grozić utratą danych.



Producent: Seikosha Ltd.

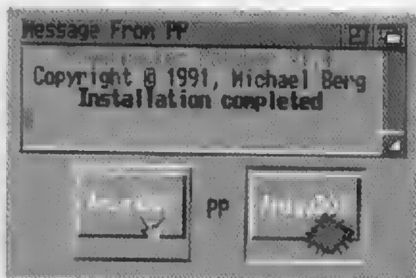
Dystrybutor: Optimus

Zgodność: Amiga, IBM
Cena: ok. 8 mln zł



Na rynku pojawiła się ostatnio kolejna drukarka matrycowa umożliwiająca dokonywanie wydruków kolorowych. Tym razem jest to drukarka 24 igłowa znanej firmy Seikosha. Jest ona bardzo szybka, ■ miarę nie duża oraz zaopatrzona standardowo w polskie znaki w systemie Mazovia i Latin-2. Jest sprzętowo zgodna ze standardem Epsona LQ 860/850. Jako nowum należy wymienić zastosowanie w niej języka drukowania ESC/P2, który pozwala na drukownie skalowanych fontów o różnych wysokościach. Maksymalna rozdzielczość drukarki to 360x360 DPI, a więc teoretycznie większa niż rozdzielczość "zwykłej" drukarki laserowej. Drukarka zapewnia również sporą szybkość wydruku, dochodzącą w porywach do 240 znaków/s (53 znaki w trybie listowym). W wersji standardowej jest zaopatrzona ■ 8 rodzajów czcionek bitmapowych i 2 czcionki skalowane (od ■ do 32 punktów).

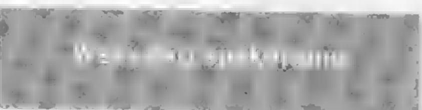
Drukarka oczywiście może współpracować również ■ taśmami jednokolorowymi, które pozwalają na zaoszczędzenie pieniędzy przy wydrukach tekstów i grafik monochromatycznych. Drukarka pracuje zarówno z pojedynczymi kartkami jak i z taśmą. Opcjonalnie może być wyposażona w podajnik papieru.



Producent: Michael Berg

Dystrybutor: Dyski PD

Zgodność: 1.2, 1.3, 2.x, 3.x
Cena: 40.000 zł



Jakiś czas temu dotarł do redakcji "PowerPacker Patcher" v1.4, który jest swego rodzaju łatką zakładaną na system. Programik jest niezwykle prosty, szybki w działaniu i skuteczny. Myślę, że warto go przypomnieć wszystkim tym, którzy jedyną nadzieję widzą w DiskEkspanderze opisywanym obok. Jeśli jakiekolwiek dane zostały spakowane przy pomocy PowerPackera, to po założeniu łaty z "Power Patchera" wystarczy je wczytać ... i zostaną one automatycznie rozpakowane. Niby nic specjalnego bo istnieje cała masa programików służących do odczytywania tak spakowanych danych (PPMore, PPPlay, itp.). Tutaj do wczytywania danych można użyć praktycznie każdego programu! Może to być CED, który nie ma zdolności ich rozpakowywania, może to być też np. DPaint, któremu przedtem trzeba było spakowany rysunek po prostu rozpakować. Teraz wszystko jest rozpakowywane automatycznie, bez żadnych specjalnych poleceń! W ten sposób możemy także spakować biblioteki (Libraries) i zostaną ■■ najpierw wczytane, a potem dostarczone systemowi. Dzięki temu prostemu programikowi wszyscy Amigowcy stali się posiadaczami o wiele pojemniejszych stacji dysków. Czasami warto przeglądać stare dyski...



























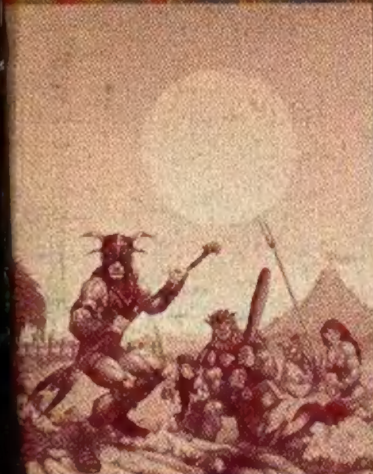

ZASMASHKA
#1
TPP/APPLAUSE©






effect

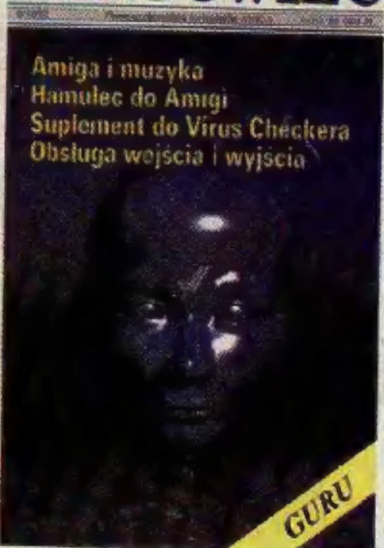

#3



















ZAMÓWIENIE

imię ....

nazwisko ....

adres ....

TEN ZESTAW MOŻE BYĆ TWÓJ ZA JEDYNE **80 tys. zł**

+ AMIGOWIEC NIESPODZIANKA

Swoje zamówienie wyślij na adres:

ALFIN
sp. z o.o.
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. 28-79-20
fax 22-64-03